International Journal of Engineering & Technology, 7 (4.19) (2018) 1041-1044

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Extraction of Meaningful Information from the Web: a Brief Survey

Santosh V. Chobe[1], Dr. Shirish S. Sane[2]

[1] Research Scholar, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, India.
Savitribai Phule Pune University, Pune
[2] HOD, Computer Engineering,
K.K.Wagh Institute of Engineering Education & Research, Nashik, India.
Savitribai Phule Pune University, Pune
**Corresponding author E-mail: sanchobe@yahoo.com*

Abstract

There is an explosive growth of information on Internet that makes extraction of relevant data from various sources, a difficult task for its users. Therefore, to transform the Web pages into databases, Information Extraction (IE) systems are needed. Relevant information in Web documents can be extracted using information extraction and presented in a structured format.
By applying information extraction techniques, information can be extracted from structured, semi-structured, and unstructured data. This paper presents some of the major information extraction tools. Here, advantages and limitations of the tools are discussed from a user's perspective.

Keywords: *Information Extraction; Web Mining; Wrapper Generation; Wrapper Induction.*

1. Introduction

Due to tremendous increase in Web, an abundant amount of information exists online. As Web information is of heterogeneous nature, only browsing and searching are used to access this information [20]. Since most of this information is only available in the form of HTML documents, collected information is not suitable for insertion in the information systems. Therefore, a considerable amount of human effort is needed to translate the collected information into a structured format that can be handled by automated systems. For effective management of Web data, relevant information needs to be extracted and translated into the structured format. Such translation of unstructured information into structured information is aimed by Information Extraction (IE) tools which identify relevant information. Extraction rules are used for recognizing the portions of a document that contain relevant information. These extraction rules are suitable for information extraction from a Web site. These set of rules is called a wrapper [19].

2. Web Mining

It extracts information from Web hyperlink structure, Web page contents, and Web data usage. Tasks of Web mining can be grouped as - Web structure mining, Web content mining and Web usage mining [21].

2.1. Web structure mining

It finds information from hyperlinks that represent the Web structure. It aims at developing techniques to analyze hyperlinks for collective assessment of quality of Web page. It discovers link structure of the hyperlinks among the documents.

2.2. Web content mining

It discovers the useful information from contents of Web page such as text, images and audio/video files. It focuses on techniques for helping a user to find contents that meet certain criteria [21].

2.3. Web usage mining

It finds access patterns based on server logs which record navigation of each user from every click made by each user [21]. The user behavior is predicted while user is interacting with the Web [13].

3. Web Information Extraction Tools Taxonomy

Taxonomy presented here is based on the techniques considered by these tools to build a wrapper. These tools are classified in the following groups: HTML aware tools, NLP based tools, Wrapper Induction tools, Modelling-based tools, and Ontology-based tools. Here, main features of these tools from each group are described.

3.1. HTML aware Tools

These tools use structural features of HTML documents to achieve extraction of data. HTML documents are converted into a parse tree, and then the extraction process is performed. Then extraction rules generation is performed semi-automatically and these rules

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

are applied to the tree. XWRAP [12], and RoadRunner [15] are some of the tools based on this approach.

3.2. NLP based Tools

These tools use NLP for learning extraction rules to extract relevant data present in Web pages. The techniques applied by such tools are filtering, syntactic and sematic tagging for building the relationships. The Web pages those consist of unstructured data, such as advertisements, can use NLP-based tools more appropriately. RAPIER [5], SRV [6], and WHISK [7] are the tools that use this approach.

3.3. Wrapper Induction Tools

These tools do not depend on linguistic constraints instead pay more attention to formatting features. Due to this, these tools are more appropriate for Web pages than previous tools. WIEN [1], SoftMealy [3], STALKER [8] and CTVS [22] are the tools that use this approach.

3.4. Modelling-based Tools

The tools in this category work in the following manner – a target structure for objects of interest is given, and then the tools locate portions of data in Web pages that follow the given structure. The structure is furnished in keeping with modeling primitives that follow an inherent data model. Tools such as NoDoSE [4] and DEByE [10, 17] adopt this approach.

3.5. Ontology-based Tools

All previous approaches consider structure of Web page. These tools rely directly on data for achieving data extraction. Ontology can be used to discover data that exists in the Web page and to construct objects with them in a given domain. The work of the Data Extraction Group at Brigham Young University represents ontology-based approach [9].

4. An Overview of Web Information Extraction Tools

4.1. HTML aware Tools

XWRAP – It is a tool that constructs wrappers semi-automatically [12]. The task of wrapper building is made easy by providing user friendly interface. There are two phases in wrapper generation process: first, *structure analysis*, and the second, *source-specific XML generation.* First the bad HTML tags are cleaned up, syntactical errors are removed, then a parse tree is generated from the document and finally, the extraction process is accomplished. XWRAP tool then interacts with users through different extraction steps to identify regions, useful hierarchical structures of the page and semantic tokens. Finally, a wrapper for a specific source is given as an output. Further, within the object extraction step, predefined data extraction heuristics are deployed that are designed for HTML pages. User should understand the HTML parse tree, also should be able to identify the tags that separate rows and columns in a table. Therefore this system requires special expertise of users [20]. Table 1 shows experimental results followed by discussion.

Table 1: Performance of Wrappers

Data Source	Fetch Time (ms)	Expand Tree Time (ms)	Extraction Time (ms)	Generation Time (ms)	Total Time (ms)	Correlation Doc/ Time
NOAA	4391	8531	3841	1128	18520	0.45
CIA Fact Book	1907	11916	4709	3902	23043	0.93
Buy.com	6908	7777	2748	838	18909	0.66
Stockmaster	1972	5489	1412	468	9973	0.35

Table 1 shows the execution time of wrappers. It can be observed that the time taken to process form-oriented pages (e.g. NOAA, Stockmaster) is almost the same. The correlation between input document size and total elapsed processing time, for variable-sized pages in Buy.com and CIA Fact Book, has been computed. For Buy.com correlation is 0.66 and for CIA Fact Book it is 0.93. Higher value of correlation indicates high selectivity i.e. same input and output size for CIA Fact Book, and lower value of correlation indicates lower selectivity i. e. input is almost 10 times the output size for Buy.com. It indicates that the wrappers are performing consistently [12].

W4F – It is a tool for generating wrappers [16]. In this tool, the process of wrapper development consists of three phases: the description about accessing the document is given by user in the first phase. The pieces of data to be extracted by user are described in the second phase and in the third phase, user declares which structure to be used for storing extracted data. A page is accessed from the Web, cleaned and then input to a parser for construction of a parse tree that follows Document Object Model (DOM) [23]. Then data is located in the parse tree using extraction rules. The extracted data is stored into Nested String List (NSL), the internal format of W4F. HTML Extraction Language (HEL) is used by W4F to express extraction rules that uses path in the HTML parse tree to address the data to be located. A given Web document, after the annotation with additional information is presented to the user [18].

4.2. NLP based Tools

SRV - It is an information extraction algorithm that works in top-down manner [6]. In [14], Freitag proposes an SRV system that takes tagged documents as input and some features are extracted those describe the tokens which can be extracted from a document. It relies on token-oriented features. These features are classified as *simple* or *relational.* A function mapping a token to some discrete value is simple feature whereas a token is mapped to another token by relational feature. SRV could extract from HTML pages because of the existence of HTML-specific features in its feature set.

WHISK – It is a tool that extracts data from text documents [7]. Given a training set of pages, WHISK generates regular expressions that are used to recognize the context of relevant instances (i.e. sentences) and the delimiters of such instances. The extraction rules are created by using tagged training instances and then the accuracy of the proposed rules is tested. In WHISK, the rules are induced in top-down fashion. They are started from the most general rule covering all instances, and then added one term at a time to extend it further. WHISK is multi-slot i.e., several records can be extracted from a single document.

4.3. Wrapper Induction Tools

STALKER –It deals with hierarchical data extraction [8]. A concept of Embedded Catalog Tree (ECT) is introduced that describes the structure of pages. ECT represents the structure of a page as a tree. The list of k-tuples is represented by the internal nodes of the ECT, in which each item in the k-tuple can be either a leaf l or another list L, called embedded list. In ECT, the complete sequence of tokens is represented by the root and each internal node is associated with portions of the sequence and each leaf node is an attributes to be extracted. STALKER is appropriate to extract

data from hierarchically structured data sources [19]. The wrapper requires a rule to extract every node in the tree from its parent. Additionally, for every list node, the wrapper needs a listing iteration rule for obtaining individual tuples by decomposing the list [20].

ShopBot is a domain-independent comparison-shopping agent [2]. It is devoted to extracting information from pages related to Web services [19]. ShopBot operates in two phases: first, learning phase and second, online comparison-shopping phase. Learning phase is performed offline. During this phase, ShopBot generates symbolic vendor descriptions of each site. Together with the domain description, this knowledge is used in comparison-shopping phase for finding products at this vendor. In first phase some simple heuristics are used to identify an appropriate search form, and to determine how to fill in the form. Then the learner must distinguish the format of product descriptions from the result page. The learning algorithm is capable of getting the format of descriptions of the products, but it cannot get the labels of the information slots. In second phase, information is extracted from the sites by using learned vendor descriptions and the best price is found for the product specified by the user [11].

CTVS – This approach extracts data automatically from query result pages [22]. CTVS follows two-steps for records extraction from a query result page.

1. Record extraction identifies the query result records (QRRs) in query result page. It has two sub steps: first, data region identification and second is segmentation.

Record extraction takes place as per the steps discussed next. First of all, in the *Tag Tree Construction* step, a tag tree is built for a query result page. In the tag tree, each node represents a tag in the HTML page and the tags enclosed inside it are its children. All possible data regions contain dynamically generated data and these are identified by the *Data Region Identification* module. As per the tag patterns, identified data regions are then segmented into data records by *Record Segmentation* module. Then the data regions that contain similar records are merged by *Data Region Merge* module from the segmented data records. Finally, one of the merged data regions is selected randomly as the one that contains QRRs by *Query Result Section Identification* module

2. *Record alignment* - The data values for the same attribute need to be aligned into the same column in the table, so *Record alignment* aligns the data values of the QRRs into a table.

Record alignment takes place in three steps explained as below -

i. *Pairwise QRR alignment* – In this step the data values are aligned in a pair of QRRs.

ii. *Holistic alignment* - In this step the data values present in all QRRs are aligned.

iii. *Nested structure processing* - In this step the nested structures that are present in QRRs are identified.

4.4. Modelling-based Tools

NoDoSE - It determines the structure of documents semi-automatically and extracts data from these documents. The user can hierarchically decompose semi structured documents [4]. The documents are decomposed in different levels. For each level of decomposition, first, the user builds an object with the complex structure, and then this object is decomposed in other objects having a more simple structure. This is accomplished by a mining component. The two heuristic-based mining components are used to derive the grammar of input documents. One component is used to mine text files whereas the other parses HTML code.

DEByE - It is a tool for wrapper generation that receives example objects as input taken from a Web document [10, 17]. Then extraction patterns are generated that permit extraction of new objects from other similar documents. The novelty of the tool is in the fact that examples are specified by user according to a structure of his liking and that this structure is described at example specification time. The patterns that allow extracting data from new documents are then generated by using examples provided by the user.

Table 2: Comparison of Different Tools

Tools	Tool Type	User Expertise	Features Used
XWRAP	HTML-aware	Programming	DOM tree
W4F	HTML-aware	Programming	DOM tree path addressing
SRV	NLP-based	Labelling	Syntactic/Semantic
WHISK	NLP-based	Labelling	Syntactic/Semantic
STALKER	Wrapper Induction	Labelling	HTML tags
ShopBot	Wrapper Induction	Labelling	HTML tags
CTVS	Wrapper Induction	No manual intervention required	DOM tree
NoDoSE	Modelling-based	Labelling	HTML tags
DEByE	Modelling-based	Labelling	HTML tags

5. Conclusion

In this paper we have presented major information extraction tools for building wrappers for information extraction from Web documents. In these tools, the focus is on simplifying the wrapper building task that is traditionally achieved by writing code in different programming languages like Java. We have considered the type of technique used by these tools for building wrappers while presenting the taxonomy of these tools. From Table 2, it can be observed that CTVS requires no user expertise as it extracts the information automatically. However, it works on some assumptions like each QRR page will have at least two QRRs. Therefore, CTVS cannot extract the information from the pages that contain single QRR. Hence, there is a need to have a tool which can extract information from the pages that contain single QRRs also.

References

[1] Kushmerick, N., Weld, D., and Doorenbos, R., "Wrapper Induction for Information Extraction," Proceedings of the Fifteenth International Conference on Artificial Intelligence (IJCAI), 1997, pp. 729-735.

[2] Doorenbos, Robert B., Oren Etzioni, and Daniel S. Weld, "A Scalable Comparison-Shopping Agent for the World-Wide Web," Proceedings Of The First International Conference On Autonomous Agents, ACM, 1997, pp. 39-48.

[3] Hsu, C.-N. and Dung, M., "Generating Finite-State Transducers For Semi-Structured Data Extraction From The Web," Journal of Information Systems, vol. 23, no. 8, 1998, pp. 521-538.

[4] Adelberg, B., "NoDoSE - A Tool For Semi-Automatically Extracting Structured And Semi-Structured Data from Text Document," SIGMOD Record, vol. 27, no. 2, 1998, pp. 283-294.

[5] Califf, M. and Mooney, R., "Relational Learning Of Pattern-Match Rules For Information Extraction," Proceedings of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing Stanford, California, March, 1998.

[6] Freitag, D., "Information Extraction From HTML: Application Of A General Learning Approach," Proceedings of the Fifteenth Conference on Artificial Intelligence (AAAI-98).

[7] Soderland, S., "Learning Information Extraction Rules For Semi-Structured And Free Text," Journal of Machine Learning, vol. 34, no. 1-3, 1999, pp. 233-272.

[8] Muslea, I., Minton, S., and Knoblock, C., "A Hierarchical Approach to Wrapper Induction," Proceedings of the Third International Conference on Autonomous Agents (AA-99), ACM, 1999, pp. 190-197.

[9] Embley, David W., Douglas M. Campbell, Yuan S. Jiang, Stephen W. Liddle, Deryle W. Lonsdale, Y-K. Ng, and Randy D. Smith., "Conceptual-Model-Based Data Extraction from Multiple-Record

Web Pages," Data & Knowledge Engineering 31, no. 3, 1999, pp. 227-251.
[10] Ribeiro-Neto, B., A., Laender, A., H., F. and DA Silva, A., S., "Extracting Semi-Structured Data Through Examples," Proceedings of the Eighth ACM International Conference on Information and Knowledge Management (CIKM), Kansas City, Missouri, 1999, pp. 94-101.
[11] Eikvil, Line. "Information Extraction from World Wide Web - A Survey," 1999.
[12] Liu, L., Pu, C., and Han, W., "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.
[13] Kosala, Raymond, and Hendrik Blockeel., "Web Mining Research: A Survey," ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, 2000, pp. 1-15.
[14] Freitag, Dayne, "Machine Learning For Information Extraction In Informal Domains," Machine Learning, vol. 39, no. 2-3, 2000, pp. 169-202.
[15] Crescenzi, V., Mecca, G. and Merialdo, P., "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," Proceedings of the 26th International Conference on Very Large Database Systems (VLDB), Rome, Italy, 2001, pp. 109-118.
[16] Sahuguet, Arnaud, and Fabien Azavant, "Building Intelligent Web Applications Using Lightweight Wrappers," Data & Knowledge Engineering, vol. 36, no. 3, 2001, pp. 283-316.
[17] Laender, A. H. F., Ribeiro-Neto, B. and DA Silva, A., S., "DEByE -Data Extraction by Example," Data and Knowledge Engineering, vol. 40, no. 2, 2002, pp. 121-154.
[18] Laender, Alberto HF, Berthier A. Ribeiro-Neto, Altigran S. Da Silva, and Juliana S. Teixeira, "A Brief Survey Of Web Data Extraction Tools," ACM SIGMOD Record, vol. 31, no. 2, 2002, pp. 84-93.
[19] Flesca, S., Manco, G., Masciari, E., Rende, E., & Tagarelli, A., "Web Wrapper Induction: A Brief Survey," AI Communications, vol. 17, no. 2, 2004, pp. 57-61.
[20] Chang, C. H., Kayed, M., Girgis, M. R., & Shaalan, K. F., "A Survey Of Web Information Extraction Systems," IEEE Transactions On Knowledge And Data Engineering, vol. 18, no. 10, 2006, pp. 1411-1428.
[21] Liu, Bing., "Web Data Mining: Exploring Hyperlinks, Contents, And Usage Data," Springer Science & Business Media, 2007.
[22] W. Su, J. Wang, F. H. Lochovsky, and Yi Liu, "Combining Tag and Value Similarity for Data Extraction and Alignment," IEEE Transactions Knowledge and Data Engineering, vol. 24, no. 7, July, 2012, pp.1186-1200.
[23] WORLD WIDE WEB CONSORTIUM. W3C. The Document Object Model. https://www.w3.org/DOM/.