International Journal of Emerging Engineering Research and Technology
Volume 2, Issue 3, June 2014, PP 186-193
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Complex Event Processing Based User-Friendly Web Data Extraction

Dr.V.Govindasamy[1], Hariharan.S[2], Hariharan.R[2] and G.Suresh[2]

[1]Department of Information Technology, Pondicherry Engineering College, Puducherry, India
(Assistant Professor, Department of IT)
[2]Department of Information Technology, Pondicherry Engineering College, Puducherry, India
(Final Year, B.Tech)

Abstract: *Internet presents a huge amount of useful information which is usually formatted for its users, which makes it difficult to extract relevant data from various sources. The availability of robust Information Extraction (IE) systems that transform the web pages into program-friendly structures such as a Relational Database has become a great necessity. Although, many approaches for data extraction from web pages have been developed, there has been limited effort to compares such tools. The data on the web is highly unstructured. There is a need some algorithm that can generate the structure from this unstructured data automatically without any manual intervention. The proposed tool is more efficient then the state of the art tools available for web data extraction. In this paper, a general purpose Web Data Extractor called as WDE, that performs well in both rigidly and loosely structured records in an HTML document is described. Our tool has been evaluated on several kinds of web pages, including product listings, search engine results pages, sports scoreboards, forums, and blogs. A Framework incorporating WDE and Complex Event processing may be evolved to process online advertisement and the interested target audience.*

Keywords: *Web Data Information Extraction, Wrappers and Complex Event Processing,*

1. INTRODUCTION

There is a lot of useful information present in Internet. This information is unstructured. There is a need for automatic information extraction from web pages. There are multiple such tools available in market. Due to the heterogeneity and the lack of structure of Web information sources, access to this huge collection of information has been limited to browsing and searching. Some algorithm that can generate the user-friendly structure from this unstructured data automatically without any manual intervention is needed. A search engine returned result page may contain search results that are organized into multiple dynamically generated sections in response to a user query.. A recent survey reveals that there are hundreds of thousands of search engines on the Web. Many web applications, such as meta search engines, deep web crawlers and shopping agents, need to interact with search engines. There is a great demand to develop automated tools (wrappers) to extract Search Result Records (SRRs) from the HTML result pages returned by search engine. Complex Event Processing (CEP) is a method of tracking and analyzing data streams about things that happen[1].

2. LITERATURE SURVEY

Programs that perform the task of IE are referred to as extractors or wrappers. In an information extraction system, a wrapper is generally a program that "wraps "an information source. Wrapper Induction (WI) or information extraction (IE) systems are software tools that are designed to generate wrappers [2].

Structured data objects are often records from underlying databases and displayed in Web pages with some fixed templates. In this paper, it is called as data records. Mining data records in Web pages is useful because they typically present their host pages' essential information, such as lists of products and services. Wrappers are specialized program routines that automatically extract data from Internet

websites and convert the information into a structured format[3].

According to [4], over 80% of the published information on the WWW is based on databases running in the background. When compiling this data into HTML documents, the structure of the underlying databases is completely lost. Wrappers try to reverse this process by restoring the information to a structured format. With the right programs, it is even possible to use the WWW as a large database. By using several wrappers to extract data from the various information sources of the WWW, the retrieved data can be made available in an appropriately structured format.

In the past few years, many approaches to WI systems [5] including machine learning and pattern matching techniques have been proposed, with various degree of automation. Earlier systems are designed to facilitate programmers in writing extraction rules, while later systems introduce machine learning for automatic rule generalization. Therefore, the user interaction has evolved from writing extraction rules to labeling target extraction data. In recent years, more efforts are devoted to reducing labeling and creating WI systems with unlabelled training examples.

2.1 Drawbacks of the Existing System

One area of previous research related to this work is wrapper generation. A wrapper usually performs a pattern matching procedure which relies on set of extraction rules. This approach is not scalable to large number of pages. This method requires still substantial manual efforts.

There are two main approaches to wrapper generation. The first approach is wrapper induction, which uses supervised learning to learn data extraction rules from a set of manually labeled positive and negative examples. Manual labeling of data is, however, labor intensive and time consuming. Additionally, for different sites or even pages in the same site, the manual labeling process needs to be repeated because they follow different templates/schemas. Example wrapper induction systems include WIEN [6], Softmealy [7], Stalker [8], WL2 [9], [10] and etc.

The second approach is automatic extraction. The method is based on a set of heuristic rules, e.g., highest-count tags, repeating-tags and ontology-matching. It proposes a few more heuristics to perform the task without using domain ontology. However, these methods produce poor results. In addition, these methods do not extract data from data records. In [10], a study is made to automatically identify data records boundaries based on a set of heuristic rules and domain ontology. Domain ontology is costly to build [11]. In [12], some additional heuristics are proposed to perform the task without using domain ontology. However, the experiment result in [13] shows that the performance of this approach is not satisfactory. In [14], a method (Iepad) is proposed to find patterns from the HTML tag string of a page and then use the patterns to extract data items. The method uses the Patricia tree and sequence alignment to find inexact matches. The problem with Patricia tree is that it is only able to find exact matches of patterns. In the context of the Web, data records are seldom exactly the same. Thus, a heuristic method based on string alignment is also proposed in [15] to find inexact matches.

In [16] a method that tries to explore the detailed information page behind the current page to extract data records is proposed. The need for detailed information pages behind is also a serious limitation. Furthermore, the method assumes that the detail pages are given, which is not realistic in practice. Due to a large number of links in a typical Web page, automatically identifying links that point to detailed information pages is a non-trivial task. A string matching method is proposed in [17]. However, its results are weak as shown in [13].

Another assumption that most current systems [18] make is that the relevant information of a data

record is contained in a contiguous segment of the HTML code. However, in some Web pages, the description of one object may intertwine with the descriptions of some other objects. For example, the descriptions of two objects in the HTML source may follow this sequence, part1 of object1, part1 of object2, part2 of object1, part2 of object2. Thus, the descriptions of both object1 and object2 are not contiguous. However, when they are displayed on a browser, they appear contiguous to human viewers.

3. PROPOSED SYSTEM

Most of the previous tools are tailored to specific web sites. The MDR [13] and Depta [18] approaches aim at extracting product listings or data displayed in tabular form. A major shortcoming of these works is that they use tree-edit distance strictly. Doing so works well on strictly structured records, such as product descriptions, but not so well on loosely structured records, such as blog entries.

In this paper, A general purpose Web Data Extractor called WDE, that performs well in both rigidly and loosely structured records in an HTML document is described.

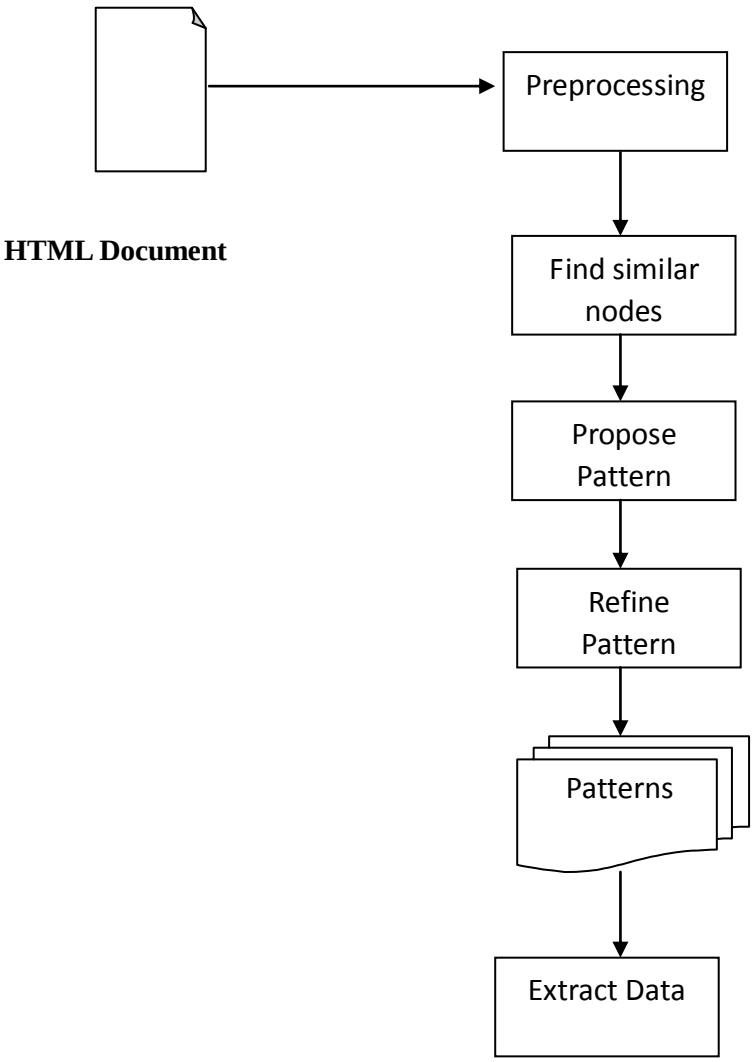

Figure 3.1 *Overview of the Proposed Work.*

Given the URL of a web page, WDE automatically discovers all repeating patterns found in the page, as follows. Initially, a standard tool for *tidying* the HTML page1 is used. The preprocessing step also merges all non-structural HTML tags (e.g., those specifying fonts, colors, etc). The result of this step

is a tree representation of the web page. The next step is identifying clusters of tree nodes with similar structure. As in previous work, the similarity between two nodes is computed in the tree by a tree matching algorithm that approximates the true tree-edit distance value between those nodes. In the third step, all possible *candidate* records in the page are listed. Candidate records can have one or more nodes, but cannot have more than one node from the same cluster. Also, it is assumed that records do not overlap. The fourth step computes the similarity of all pairs of records identified in the previous step. This is done as follows. Each record is converted into a tree whose root is a *dummy* node containing all nodes in that record as children. The similarity between two records is computed similarly as in step 2. Finally, the records based on their similarity are clustered. From the clusters obtained after step 4, the data extraction patterns that navigate the HTML tree are generated.

4. SYSTEM DESIGN

The system architecture is designed in such a way that input HTML document is parsed and DOM tree view is provided to the user and the specified data is extracted.

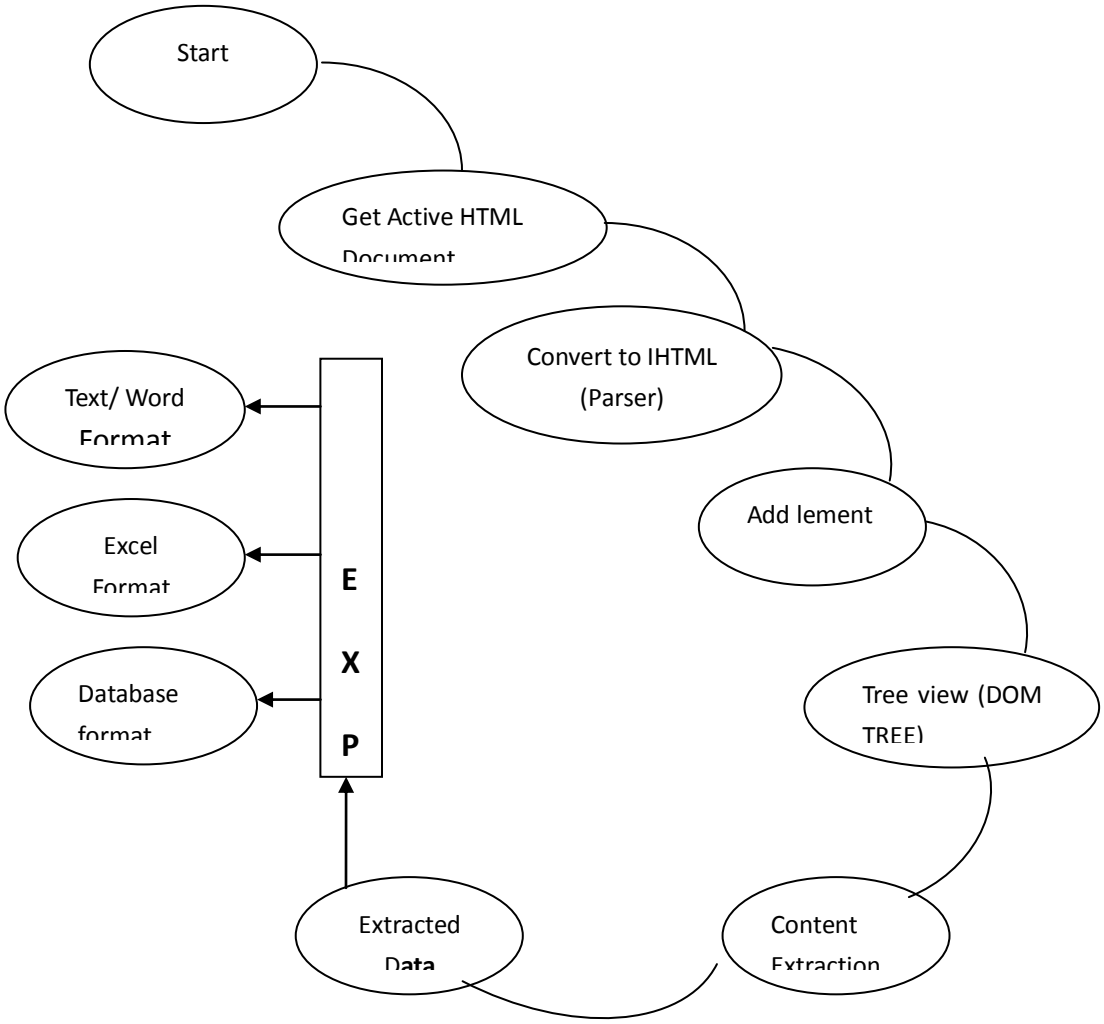

Figure 4.1 *Data Flow Diagram for WDE*

4.1 Input Description

The user inputs the HTML document to be extracted. The system first parses the HTML document and gives the DOM tree view to the user. From the tree view user identifies the node which contains information to be extracted. The input to the tool is set of HTML documents in the way described below.

4.1.1 System Inputs

The system inputs are HTML documents with extensions such as .htm, .html. The tool first it checks for the appropriate extension and then it loads the document for parsing.

4.1.2 Input Interface

The tool after parsing the HTML document which is taken into active form gives the DOM tree view. The DOM tree view provides the appropriate interface for the user to extract the specified information.

4.2 Output Description

The requested data of the client will be extracted and displayed in list box. This helps the user for further refining the extraction procedure.

4.2.1 System Outputs

The output of the system is the requested data of the user. The extracted information can be exported to different formats as required by the user. The extracted content can be viewed in terms of text file, excel sheet or Database format.

4.2.2 Output Interface

The necessary buttons provided in the tool helps the user in extraction procedure. The user has the option of storing the content in different formats. File storage and location information also provided to the user in guiding them further.

4.3 Algorithm

1. Start
2. Get Active document
3. Create IHtml Doc (DOM file)
4. Set Tree ID as 1
5. Add Element which calls add child proc
6. if Relative element is NULL then consider it as root element and place on the root tree
7. From the DOM tree view perform the narrow band extraction method as follows:
   a. If length > HBegin && length < HEnd then extract value.
   b. if child_ID >= ChildBegin And child_ID <= ChildEnd
   c. Then selected/marked text is filled into list
8. Then Export the List /extracted content to File or Database.
9. end

5. EMPIRICAL EVALUATIONS

This section evaluates the proposed system, WDE implements the proposed techniques. Multiple experiments are conducted on pages collected from 22 websites. These sites are used by other systems (RISE, OMINI, IEPAD, ROADRUNNER, and EXALG) and some well-known sites from a wide range of domains. The system is fully automatic without human intervention. The tool has been compared with the MDR (Mining Data Records) and DEPTA (Data Extraction based on Partial Tree Alignment) method, which is the state-of-the-art in web data extraction based on tree edit distance. The input is a single page and the output is (i) A new text file that contains a list of data records. Each data records contain a list of data items in the input page. (ii) An Excel file that contains a list of data records.

5.1 Experimental Result

The standard parameters of precision and recall [19] are used to evaluate the tool. Recall is defined as

the percentage of the intended data records that are retrieved by the tool; precision is defined as the percentage of the returned data records that are correct. The correctness (i.e., precision) manually, on a best-effort basis are determined.

Precision measures the number of correctly identified items as a percentage of the number of items identified. In other words, it measures how many of the items that the system identified were actually correct, regardless of whether it also failed to retrieve correct items. The higher the Precision, the better the system is at ensuring that what has been identified is correct.

It is formally defined as:

$$\Pr ecision = (Correct + Mis\sin gDR)/TotalData\,\mathrm{Re}cords$$

Recall measures the number of correctly identified items as a percentage of the total number of correct items. In other words, it measures how many of the items that should have been identified actually were identified, regardless of how much spurious identification was made.

The higher the Recall rate, the better the system is at not missing correct items. Recall is formally defined as:

$$\mathrm{Re}call = CorrectDR/TotalData\,\mathrm{Re}cords$$

It is shown in Table 5.1 the performance of WDE on data record extraction and data item alignment. Column A denotes the numbers of actual data records. For comparison purpose, these numbers are manually counted. Note that only count the data records which contain valued information. Some area such as navigation bars in a page also contains data with regular patterns and they will be identified by our proposed technique. WDE takes into consideration the visual information of identified data records to decide whether to output them or not. For example, if a list of data records locate on the boundary part of a page and each occupies a comparatively small area, they are considered as unimportant data and will not be outputted.The criteria used to decide the importance of a list of data records is left to the user based on the range set by him. Columns C denotes the numbers of correct data records that are extracted by WDE. Column W denotes the numbers of data records and data items extracted/aligned incorrectly by WDE.

For a data record, incorrect extraction usually has the following cases:

1. Only part of the content of the data record is extracted;

2. Information outside of the data record boundary are extracted and enclosed in it.

Column S denotes the data records that are retrieved by the tool are spurious. In other words it specifies the repetition of data records. Column M denotes the number of data records that were not identified by WDE. The last three rows of the table give the total of each column, the recall and precision of the system.

Table 5.1 *Experimental Results*

Site	**A**	**C**	**W**	**M**
www.ashford.com	24	24	0	0
www.cameraworld.com	40	40	0	0
www.pricegrabber.com	12	12	0	0
www.bestbuy.com	30	28	2	0
www.amazon.com	24	24	0	0
www.overstock.com	120	120	0	0

www.buy.com	24	22	0	2
www.kithabay.com	20	20	0	0
www.ebay.com	100	96	2	2
www.spicesbookstore.com	16	16	0	0
www.abt.com	40	40	0	0
www.rochesterclothing.com	32	32	0	0
www.smartbargains.com	48	48	0	0
www.nothingbutsoftware.com	30	30	0	0
www. shopping.yahoo.com	30	30	0	0
www.nextag.com	30	30	0	0
www.bargainoutfitters.com	12	12	0	0
www. refurbdepot.com	18	16	2	0
www.dealtime.com	20	19	1	0
www.toshiba.com	32	32	0	0
www.circuitcity.com	12	12	0	0
www.compusa.com	16	16	0	0
Total	730	719	7	4
Average Recall (%)	98.49 %			
Average Precision (%)	99.04 %			

6. CONCLUSION

In this paper, A new approach for extracting semi-structured data records from web pages is presented. Although the problem has been studied by several researchers, existing techniques are either inaccurate or make more assumptions. The proposed method does not make such assumptions. It only requires that pages should not contain exactly similar data records, which is always true in reality. A personalized template mechanism is provided for the user convenience. User can save the pattern after extracting data from a specific site. This pattern makes the data extraction faster when visiting the site next time. This unique feature is not present in any of the existing tools. Evaluation results reveal that our proposed tool WDE performs more efficient than MDR and DEPTA which is state-of-art in web data extraction. Hence, the proposed tool will be very useful for web data extraction applications and more convenient to the user. There is a scope to improve this tool by removing the spurious contents in extraction. There is a possibility to make refined template to overcome such problem. Presently our system cannot automatically label the extracted data items. This problem is addressed in [20] but the solution proposed is not general enough. In [21], queries are sent out through complex search forms and the search results are used for data extraction and labeling. However, most Web sites do not provide complex. A Framework incorporating WDE and Complex Event processing can been evolved to process online advertisement and the interested target audience.

REFERENCES

[1] V. Govindasamy and P. Thambidurai, April 2013,"Complex Event Processing – A survey", Journal of Computing, Vol-101, No-1,p 125-137.

[2] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, Khaled Shaalan, 2006. "A Survey of Web Information Extraction Systems". IEEE Transactions on Knowledge and Data Engineering, TKDE0475-1104.R3.

[3] A. Arasu and H. Garcia-Molina, 2003. "Extracting structured data from web pages". In SIGMOD.

[4] D. Buttler, L. Liu, and C. Pu, 2001. "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, pp 361, Washington, DC, USA. IEEE Computer Society.

[5] C.-M. Hoffmann and M.J. O'Donnell, 1982. "Pattern matching in trees". Journal of ACM, Vol-29, No-1,pp,68-95.

[6] N. Kushmerick. Wrapper induction, 2000 "effciency and expressiveness. Artificial Intelligence", Vol-118:pp.15-68.

[7] C.-N. Hsu and M.-T. Dung, 1998. "Generating finite-state transducers for semi-structured data extraction from the web". Inf. Syst., Vol-23, No-9,pp.521-538.

[8] I. Muslea, S. Minton, and C. Knoblock, 1999 ," A hierarchical approach to wrapper induction". In AGENTS '99: Proceedings of the third annual conference on Autonomous Agents, pages pp.190-197, New York, NY, USA, ACM Press.

[9] W. W. Cohen, M. Hurst, and L. S. Jensen, 2002, "A flexible learning system for wrapping tables and lists in html documents". In WWW '02: Proceedings of the eleventh international conference on World Wide Web, pp. 232-241, New York, NY, USA ACM Press.

[10] D. Pinto, A. McCallum, X. Wei, and W.-B. Croft,2003, "Table extraction using conditional random fields". In SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp.235-242, New York, NY, USA. ACM Press.

[11] D. W. Embley, Y. Jiang, and Y. K. Ng,1999, "Record-boundary discovery in web documents". In SIGMOD.

[12] D. Buttler, L. Liu, and C. Pu, 2001, " 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, pp.361, Washington, DC, USA. IEEE Computer Society.

[13] B. Liu, R. Grossman, and Y. Zhai, 2003, "Mining data records in web pages". In KDD '03:Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 601-606, New York, NY, USA. ACM Press.

[14] C. Chang and S. Lui, 2001, "IEPAD: Information extraction based on pattern discovery". In WWW '01: Proceedings of the tenth international conference on World Wide Web.ACM Press.

[15] H. Carrillo and D. Lipman, 1988, "The multiple sequence alignment problem in biology. SIAM J.Appl. Math., Vol-48,No-5, pp.1073-1082.

[16] Lerman, K., Getoor L., Minton, S. and Knoblock, C, 2004, "Using the Structure of Web Sites for Automatic Segmentation of Tables". SIGMOD-04.pp.4

[17] Baeza-Yates, R., 1989, "Algorithms for string matching: A survey.ACM SIGIR Forum",Vol-23,No.(3-4),pp.34-58, 1989.

[18] Y. Zhai, and B. Liu, 2005, "Web Data Extraction Based on Partial Tree Alignment". In WWW '05: Proceedings of the 14th international conference on World Wide Web, pp. 76–85.

[19] Diana Maynard Wim Peters Yaoyong Li, 2006, "Metrics for Evaluation of Ontology based Information Extraction". WWW 2006, May 22–26, Edinburgh, UK.

[20] J. Wang and F. H. Lochovsky, 2003, "Data extraction and label assignment for web databases". In WWW '03: Proceedings of the twelfth international conference on World Wide Web, pp. 187-196, New York, NY, USA. ACM Press.

[21] L.Arllota, V. Crescenzi, G. Mecca, and P.Merialdo, 2003, "Automatic annotation of data extraction from large web sites". In the International Workshop on the Web and Databases, pp 7-12.