A Novel Automated Scheme For Creating Generic Wrappers For Structured Web Sources [GWSWS]

P.J. Kulkarni, Jayshree Janardan Jagtap and Anil Ramchandra Surve

Abstract-- **In today's world the World Wide Web has became the ultimate "all kind of information" repository to place and to obtain information for humans very successfully. There's an emerging need occurs, namely, going beyond the concept of "human browsing" by facilitating the automated process of information retrieval and enabling further utilization by targeted applications as per their need. So it is better and necessary to have an automatic mechanism able to retrieve the requested required data. It is possible for a user to visit a list of sites and retrieve pieces of information in an automated repeated process.**

The information and data on the Web is usually presented in the form of mostly structured HTML pages. As this structure is not known in advance, sophisticated mechanisms are needed to automatically discover and extract structured Web data. The most obvious problem in designing such a system for Web information extraction is the lack of homogeneity in the structure of the source data found in Web sites. It is mandatory for an extraction system has to present data in a structured form.

Therefore, a mechanism is needed to derive information from the diversity of structures in data sources and a dedicated piece of software is required for each Web site, to exploit this correspondence in structures. Such pieces of software are called data source wrappers or simply wrappers and their purpose is to extract the useful information from the Web data sources.

Index Terms--**Wrappers, Web Mining, Web Data Extraction, Web structure mining, information retrieval.**

I. Introduction

IT is very easy for humans to navigate through a Web site and retrieve the useful information. For a human worker this is a routine task, a time-consuming and tiresome activity. Instead, being able to have this information ready for use (e.g., by e-mail or sms) saves precious time and effort. Another scenario includes activities like data mining, which require a vast amount of available information (corpus) for statistical and training purposes. This information is very difficult—if not impossible—for a human to acquire manually. It is demand to have an automatic mechanism able to retrieve the required data. Notice that in both cases, we are dealing with a repeated process, a routine: visit a list of sites and then retrieve pieces of information from each of them. Humans do not perform well in such tasks; therefore, machines exist for such purposes. Once a program able to locate and extract the desired information has been developed, this process can be performed as often as and for as long as we want. As data available on the Web is commonly in HTML pages form which is mostly structured and the structure is not known in advance. To overcome this, automatic discovery and extraction systems needed to handle the structured Web data. This should take care the lack of homogeneity of the source Web data found in Web sites for information extraction. An extraction system is proposed here which aims to derive information from the diversity of structures in data sources.

The dedicated piece of software which is required for each Web site, to exploit this correspondence in structures is noted as 'data source wrappers' or simply wrappers which purposes to extract the useful information from the variety of Web data sources.

Wrappers can be divided in two main categories. One category is termed as "site specific", which are developed to extract information from a specific Web page or family of Web pages. They are very easy to develop, but not generic since they fail to retrieve the information from data sources with different structure or even from the same pages if their structure is changed. The second category includes "generic wrappers", which are developed to extract particular information. They can be applied to almost any page, regardless of the specific structure, but they are difficult to develop because of the great variety and lack of uniformity in the Web page structure. We refer to the data to be extracted by the wrapper with the term "structural tokens."

II. Related Work & Motivation

To look at the various attempts done in such regard, which is found that, a lot of work exists in the literature concerning the wrapping of Web data sources. Most of this work refers to

Dr. P.J.Kulkarni is H.O.D. of Department of Computer Engg., Walchand college sangli, India (e-mail: pjk_walchand@rediffmail.com)
J.J.Jagtap is M.E. student of Department of Computer Engg., Walchand college sangli, India (e-mail: jagtap.jayshree@gmail.com)
A. R. Surve is lecturer in Department of Computer Engg., Walchand college sangli, India (e-mail: a_nilsurve@rediffmail.com)

semiautomatic wrappers, but some automatic wrappers have also been proposed. The semiautomatic wrappers use several techniques based on query languages, labeling tools, grammar rules and natural language processing, and HTML tree processing.

As an example of a semiautomatic wrapper based on query language, documents are represented by a relational or object-oriented data schema and an interrogation using a declarative query language based on an exact matching and forced by the data schema structure is performed. In a similar approach, this uses wrapper induction and supervised learning. The system is typically trained by using manually labeled positive and negative data, to create data extraction rules. Lixto describes a similar approach, using different tools. It assists the user to semi automatically create wrapper programs by providing a fully visual and interactive user interface. In [7], text in no grammatical sentence fragments as well as text in tabular format is parsed into coherent text segments based on page layout cues.

Another important category of semiautomatic wrappers is based on a tree representation of the HTML page. The system proposed by [8] allows accessing of semistructured data represented in HTML interfaces.

All the systems presented so far are semiautomatic, meaning that human assistance is required at some point of their operation. However, other systems have been proposed in the literature, automating the process of information extraction. According to the approach in [9], the structure of a document is captured as a tree of nested HTML tags. The subtree containing the records of interest is then located. Omini [10] is a fully automated extraction system. It parses web pages into tree structures and performs object extraction in particular stages.

Summarizing, most of the existing work use human assisted techniques to extract rules and patterns in order to equip the generated wrapper. The disadvantage of this approach is the fact that even a minor change in the layout of the Web page can cause “de synchronization” between the wrapper and the data source.

III. Technical Work Preparation

A. Novel Concept

We present a novel fully automated scheme for creating generic wrappers for structured Web sources. The key idea is to exploit the format of the information contained in the Web pages and discover the underlying structure. As stated in [4], “although the Web is less structured than we might hope, it is less random than we might fear.”

Elements of the same type usually reside in the same section. Therefore, a key step toward retrieving the data is to discover the sections contained in the Web page and to identify the one holding the interesting information. Once the area of the page containing the structural tokens is located, our focus is to separate them. To do that, we use methods of clustering and statistics origin. Our innovation lies in exploiting clustering techniques in order to fully automate the process of information extraction. This permits building a system that can operate without human intervention and training. Automatic wrapping can be applied and it is justified by the way the content of a Web page is typically generated. The information about an item is retrieved from a local data source, e.g., a database, and is rendered on the page according to a predefined template. This automatic, template-based, procedure ensures that all the items to be extracted are displayed in a similar way.

B. Prototype Framework

We present here, a fully automatic wrapper that can extract the structural tokens from a given Web document. The proposed system is comprised of two modules which we call transformation and extraction module, respectively.

They are further subdivided into components, each one responsible for a different task. The overall procedure which extracts the information from the data source is performed in three distinct phases as in Fig. 1. During the first phase, we prepare the Web document for the extraction that will follow. This is done by inserting the HTML document into the transformation module, the components of which generate a tree. This tree corresponds to the inserted HTML document.

Given the previously generated tree, the second phase aims at discovering and segmenting the region of the tree in which the structural tokens are located. This is performed by the components of the extraction module. The third phase concludes the operation by mapping the selected tree nodes to elements in the initial HTML Web document. This mapping is carried out by the transformation module.

C. Framework Components

Fig. 1 shows the main components of a GWSWS and how the control flows among them.

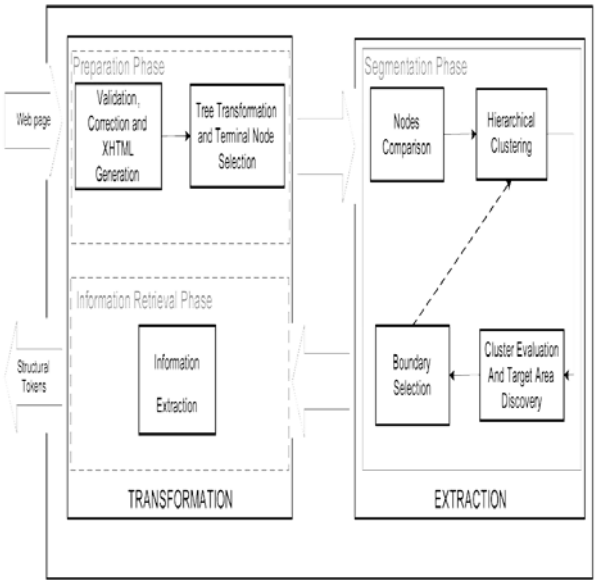

Fig. 1. Components of framework.

IV. Functionality Of Components

Phase 1: Preparation Phase

During the first phase, the Web document for the extraction will be prepared. This will be done by inserting the HTML document into the transformation module, the components of which will generate a tree. This tree will correspond to the inserted HTML document.

Phase 2: Segmentation Phase

This phase will consist of the most important and innovative part of the procedure. Given the previously generated tree, the second phase aims at discovering and segmenting the region of the tree in which the structural tokens will be located. This will be achieved by the components of the extraction module.

Phase 3: Information Retrieval Phase

At this stage, the region in the Web document that contains the structural tokens as well as the boundaries separating them will be located. Mapping the extracted nodes to the corresponding elements in the Web page, retrieving the useful information this way, will be the final step to be performed here by the component. This component, after parsing the initial Web page, will extract the desired information according to the segmentation achieved in the previous phase.

A. Phase 1: Preparation Phase

Validation, Correction, and XHTML Generation Component:

This first component performs a syntactical correction to the source's HTML by transforming it into XHTML to prepare it well formed. The cleaned and normalized page is then fed into the "Tree Transformation and Terminal Node Selection Component," which generates a tree representation of the page. The root of this tree corresponds to the whole document. The intermediate nodes represent HTML tags that determine the layout of the page. Once the tree construction is completed, the terminal nodes are selected page among which the useful information resides. The non-terminal nodes are not in our interest since they represent layout descriptive elements, in other words the way the information is displayed in a Web browser. We only select the terminal nodes for further process since they represent the elements of the We must note here that the selection of the terminal nodes happens in a way that preserves their ordering in the Web page. The ordering of the terminal nodes is critical in our application.

An example of this kind is illustrated here; the source code of a sample Web document is shown in Fig. 2 and, in Fig. 3, its corresponding tree representation is mentioned.

Fig. 2. Sample web document with its source code.

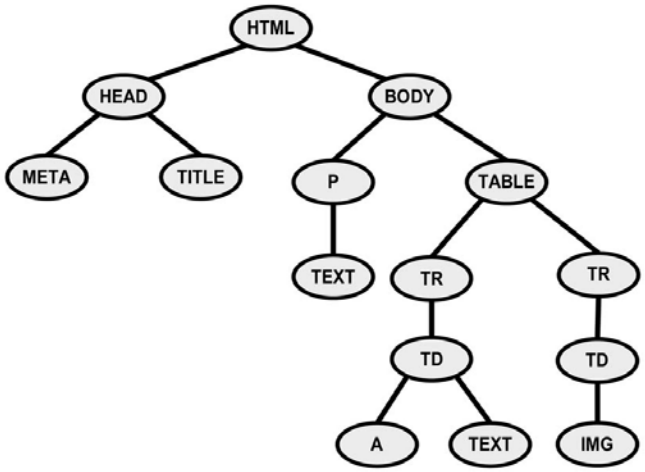

Fig. 3. The corresponding tree representation.

B. Phase I1: Segmentation Phase

This phase encapsulates the most important and innovative part of the procedure. The structural token extraction is achieved here. Recall that, at this point, we have already generated the tree representation of the Web document and also selected the terminal nodes *n1*, *n2*, . . . , *nN*, where *N* denotes the number of terminal nodes in the tree. At the present phase, we perform a segmentation of the selected terminal nodes, in a way that a one to one correspondence between the nodes representing elements of the same structural token and the extracted segments exists.

Nodes Comparison Component:

This component is responsible for calculating the N X N terminal node similarity matrix S that will be used for the clustering of these nodes.

Hierarchical Clustering Component:

This component performs a one-dimensional hierarchical clustering. Its purpose is to select a subset of the terminal

nodes generated by phase one, or equivalently, to locate a subtree in the initial tree. This subtree will be selected to correspond to the region of the Web documents containing the structural tokens.

The pseudo code in Fig. 4. illustrates the equivalence between the one-dimensional clustering of the index set T_N and the Clustering of the nodes.

```
/* Input s : a vector containing similarity measures*/
/* Input T_N : a vector the indexes to be clustered */
/* Output C_T : a vector of clusters */
  m = min s, M = max s;
  k = 0; newCluster = true;
  for each i ∈ [n, M] do {
    for each j ∈ T_N do {
      if s_j ≥ i {
        if (newCluster){
          k = k + 1; }
        insert(C_{T_{i,k}}, j);
        newCluster = false; }
      else newCluster = true;
    }
  }
```

Fig. 4. The pseudo code for the hierarchical clustering.

Cluster Evaluation and Target Area Discovery Component:

This component discovers which level in the hierarchical clustering of indices is the "cut-off" level. With the term cutoff level we mean the hierarchy depth in the cluster tree that achieves the best separability among the clusters in this level. In order to discover the cut-off level, we calculate the value that maximizes the separation criterion.

Boundary Selection Component:

At this point, we have located the region in the Web document that contains the structural tokens. The next step is to separate the structural tokens.

C. Phase II1: Information Retrieval Phase

At this point, we have successfully located the region in the Web document that contains the structural tokens as well as the boundaries separating them. Mapping the extracted nodes to the corresponding elements in the Web page, retrieving the useful information this way, is the final step performed here by the Information extraction Component.

Information Extraction Component:

This component, after parsing the initial Web page, extracts the desired information according to the segmentation achieved in the previous phase.

V. Conclusions

As per the need of a mechanism to automate the process of extraction of Web data of varied structures, we presented here a novel fully Web wrapper. The main characteristic of the proposed wrapper is the fact that, in contrast with most of the other related work considered so far, it does not require any human assistance or training phases. The main innovation and contribution of our system consists in introducing a signal-wise treatment of the tag structural hierarchy and using hierarchical clustering techniques to segment the Web pages into structural tokens. The importance of such a treatment is significant since it permits abstracting away from the raw tag-manipulating approach the other systems use. Hence such an automated Wrapper can relieve human routine and time consuming interventions to extract Web data.

VI. Acknowledgment

We wish to hereby express my deep gratitude and sincere thanks to Prof. B. F. Momin , Department of Computer Science & Engineering, Walchand college of Engineering, Sangli, India for encouraging me to go ahead with the work on " A Novel Automated Scheme For Creating Generic Wrappers For Structured Web Sources [GWSWS] "

VII. References

[1] Nikolaos K. Papadakis, Dimitrios Skoutas, Konstantinos Raftopoulos, Theodora A. Varvarigou, "STAVIES: A System for Information Extraction from Unknown Web Data Sources through Automatic Web Wrapper Generation Using Clustering Techniques" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 12, DECEMBER 2005.

[2] N. Kushmerick, "Wrapper Induction: Efficiency and Expressiveness," Artificial Intelligence, vol. 118, nos. 1-2, pp. 15-68, 2000.

[3] J. Han and K.C.-C. Chang, "Data Mining for Web Intelligence," Computer, Nov. 2002.

[4] N. Ashish and C.A. Knoblock, "Semi-Automatic Wrapper Generation for Internet Information Sources," Proc. Int'l Conf.Cooperative Information Systems, pp. 160-169, 1997.

[5] N. Kushmerick, D. Weld, and R. Doorenbos, "Wrapper Induction for Information Extraction," Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI-97), 1997.

[6] R. Baumgartner, S. Flesca, and G. Gottlob, "Visual Web Information Extraction with Lixto," The VLDB J., pp. 119-128, 2001.

[7] S. Soderland, "Learning to Extract Text-Based Information from the World Wide Web," Proc. Knowledge Discovery and Data Mining,pp. 251-254, 1997.

[8] J.-R. Gruser, L. Raschid, M.E. Vidal, and L. Bright, "Wrapper Generation for Web Accessible Data Sources," Proc. Conf. Cooperative Information Systems, pp. 14-23, 1998.

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

[10] D. Buttler, L. Liu, and C. Pu, "A Fully Automated Object Extraction System for the World Wide Web," Proc. 2001 Int'l Conf. Distributed Computing Systems (ICDCS '01), pp. 361-370, 2001.

[11] O. Etzioni, "The World-Wide Web: Quagmire or Gold Mine ?," Comm. ACM, vol. 39, no. 11, pp. 65-68, 1996.

VIII. BIOGRAPHIES

Jayshree Jagtap is graduated from the D.K.T.E Ichalkaranji. and she is currently doing her post graduation in Computer Science from the W.C.E., Sangli Maharastra, India

Her dissertation work is under “Web Mining” domain.

Anil Surve is Post graduated in Computer Science from the W.C.E., Sangli Maharastra, India and he is currently working as Lecturer in the Computer Science department.