International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 7, Issue 8 (June 2013), PP. 37-41

Review paper on "Optimized approaches for web data harvesting."

P. Singam, Prof. P. Pardhi
[1]Student M. Tech. (Comp.Sci. & Engg)
[2]Assistant Professor, Comp. Sci. & Engg. Deptt.
R.C.O.E.M., Nagpur (India)

Abstract:- Companies spend millions of dollars to build data warehouses to hold their data and data mining applications must take advantage of this. Besides saving significant manual effort and storage space, relational integration allows data mining applications to access the most up-to-date information available. To collect large data is very tedious and frustrating task. There are various kinds of valuable semantic information about real-world entities embedded in web pages and databases. Extracting and integrating these entity information from the Web is of great significance.

Keywords:- Data Mining, Data extraction, DOM structure, Web Extraction, Wrappers.

I. INTRODUCTION

In the last few years, several works in the literature have addressed the problem of data extraction from web pages. The importance of this problem derives from the fact that, once extracted, the data can be handled in a way similar to instances of a traditional database. With the explosion of the World Wide Web, a wealth of data on many different subjects has become available on line. This has opened the opportunity for users to benefit from the available data in many interesting ways. In this context automatic web data extraction plays an important role. Example of web data extraction are i) Extract competitor's price list from web page regularly to stay ahead of competition, ii) Extract data from a web page and transfer it to another application iii) Extract people's data from web page and put it in a database.

In this paper we are trying to analyze different techniques available for data extraction from web pages. There are various approaches to address the problem of web data extraction; it uses the techniques borrowed from areas such as natural language processing, languages and grammars, machine learning, information retrieval and ontologies. For handling web data more effectively, some ideas can be taken from the database area. Traditional database techniques cannot be directly applied to Web data since it require structure data. Indeed, the volume of unstructured or semi structured data available on web is enormous and still increasing. Thus, to address this problem, a possible strategy is to extract data from Web sources to populate databases for further handling.

In this paper we are presenting the literature survey which is being carried out to develop web harvesting tool for web data extraction. Section II discusses the various approaches and techniques carried out by different researchers, section III gives the overview of our approach for web data extraction, and section IV concludes the paper.

II. LITERATURE REVIEW AND DISCUSSION

Jun Kong, Omer Barkol, et al. [1] Distinct from heuristic approaches, this approach formalizes a common Web pattern as a graph grammar, which formally and visually specifies the information organization underlying a Web page. The grammar-based approach interprets a Web page from bottom to top. It needs to first recognize atomic information objects before page segmentation. Based on above assumption, this paper proposes a novel approach to page segmentation, taking advantage of graph grammars to provide robust page segmentation the spatial graph grammar (SGG) is used in this approach to analyze Web interfaces. Spatial specifications in the abstract syntax enable designers to model interface semantics with various visual effects (e.g.,a topological relation between two interface objects). This approach interprets a Web page, or any interface page, directly from its image Image-processing techniques are used to divide an interface image into different regions and recognize and classify atomic interface objects, such as texts, buttons, etc., in each region. The object recognition produces a spatial graph, in which nodes represent recognized atomic objects and edges indicate some significant spatial relationships. Finally, the SGG parser parses the spatial graph to discover the hierarchical relations among those interface objects based on a predefined graph grammar and recognizes the objects to be extracted.

Mohammed Kayed and Chia-Hui Chang [2]

In this paper, the data extraction problem has formulated as the decoding process of page generation based on structured data and tree templates. Author propose an unsupervised, page-level data extraction approach to deduce the schema and templates for each individual Deep Website, which contains either singleton or multiple data records in one Webpage. Authors schema called FiVaTech, applies tree matching, tree alignment, and mining techniques to achieve the challenging task. In experiments, FiVaTech has much higher precision than EXALG and is comparable with other record-level extraction systems like ViPER and MSE. In this paper, they focuses on page-level extraction tasks and propose a new approach, called FiVaTech, to automatically detect the schema of a Website. The proposed technique presents a new structure, called fixed/variant pattern tree, a tree that carries all of the required information needed to identify the template and detect the data schema. They combine several techniques: alignment, pattern mining, as well as the idea of tree templates to solve the much difficult problem of page-level template construction. In experiments, FiVaTech has much higher precision than EXALG, one of the few page-level extraction systems. This paper, proposes a new Web data extraction approach, called FiVaTech to the problem of page-level data extraction. They formulate the page generation model using an encoding scheme based on tree templates and schema, which organize data by their parent node in the DOM trees. FiVaTech contains two phases: phase I is merging input DOM trees to construct the fixed/variant pattern tree and phase II is schema and template detection based on the pattern tree. The proposed page generation model with tree-based template matches the nature of the Webpages. Meanwhile, the merged pattern tree gives very good result for schema and template deduction. For efficiency, they only use two or three pages as input.

Jer Lang Hong [3]

Author’s investigations indicate that the development of a lightweight ontological technique using existing lexical database for English (WordNet) is able to check the similarity of data records and detect the correct data region with higher precision using the semantic properties of these data records. The advantages of this method are that it can extract three types of data records, namely, single-section data records, multiple-section data records, and loosely structured data records, and it also provides options for aligning iterative and disjunctive data items. Tests also show that the wrapper is able to extract data records from multilingual web pages and that it is domain independent. According to the literature there are two types of data in the deep webs that can be extracted using wrappers. The first group is the list page, where a list of data records is generated from a search query and displayed as search results. The second type is a detailed page, where specific information for a product is generated for the user. This paper mainly focuses on the extraction of data records from the list page. The literature has concluded with the development of a wrapper for the extraction and alignment of data records using lightweight ontological technique. This proposed ontological technique could extract data records with varying structures effectively. The wrapper is able to distinguish data regions based on the semantic properties of data records but not the DOM tree structure and visual properties.

Weifeng Su, Jiying Wang, Frederick H. Lochovsky [4] They present a novel data extraction and alignment method called CTVS that combines both tag and value similarity. CTVS automatically extracts data from query result pages by first identifying and segmenting the query result records (QRRs) in the query result pages and then aligning the segmented QRRs into a table, in which the data values from the same attribute are put into the same column. Experimental results show that CTVS achieves high precision and outperforms existing state-of-the-art data extraction methods. In this article author employ the following two-step method, called Combining Tag and Value Similarity (CTVS), to extract the QRRs from a query result page p.

1. Record extraction identifies the QRRs in p and involves two sub steps: data region2 identification and the actual segmentation step.
2. Record alignment aligns the data values of the QRRs in p into a table so that data values for the same attribute are aligned into the same table column.

Zaiqing Nie, Ji-Rong Wen, and Wei-Ying Ma [5] In this paper they introduce the webpage understanding problem which consists of three subtasks: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling. The problem is motivated by the search applications they have been working on including Microsoft Academic Search, Windows Live Product Search and Renlifang Entity Relationship Search. They believe that integrated webpage understanding will be an important direction for future research in Web mining. They segmented a webpage into semantic blocks and label the importance values of the blocks using a block importance model. Then the semantic blocks, along with their importance values, are used to build block-based Web search engines. These entities and their relationships are automatically mined from the text content on the Web (more than 1 billion Chinese webpages).

Luis Tari, Phan Huy Tu, Jo¨ rg Hakenberg, et al., [6] In this paper author take into consideration the drawback of the existing system such that whenever a new extraction goal emerges or a module is improved, extraction has to be reapplied from scratch to the entire text corpus even though only a small part of the corpus might be affected. This paper describe a novel approach for information extraction in which extraction needs are expressed in the form of database queries, which are evaluated and optimized by database systems. For information extraction using database queries enables generic extraction and minimizes reprocessing of data by performing incremental extraction to identify which part of the data is affected by the change of components or goals. It is seen in experiments that in the event of deployment of a new module, incremental extraction approach reduces the processing time by 89.64 percent as compared to a raditional pipeline approach.

Hassan A. Sleiman and Rafael Corchuelo (7) In this article, author caries out survey the existing proposals regarding region extractors and compare them side by side. Web information extraction is the task of identifying, extracting, and structuring relevant information from web documents in structured formats, e.g., tables or XML. Author have surveyed all of the proposals on region extractors carried out within a four-dimensional comparison framework and concluded that: 1) more effort is required regarding free-text web documents since the proposals they have

surveyed rely almost exclusively on finding regular structures in a web document; 2) the majority of proposals seem scalable since they are unsupervised; 3) there are no conclusive and comparable results regarding their efficiency and effectiveness; and 4) none of them is universally applicable.

The importance of data regions extractors is clear, some of the proposals they have analysed are now an intrinsic part of recent information extractors or have inspired them; there are also applications to fields such as information retrieval, focused web crawling, topic distillation, adaptive content delivery, mashups, and metasearch engines. In this article, author have surveyed region extractors and compared them regarding four dimensions, namely: input and output, algorithmic, efficiency and effectiveness and some miscellaneous features. The following conclusion can be drawn from the survey:1) All of the region extractors work on semi-structured documents that are formatted in HTML and rely on their DOM tree directly or indirectly by using VIPS. In general, they search for repetitive structures to identify data regions. This makes it difficult to apply them to free-text documents whose contents do not rely heavily on HTML tags.

2) The majority of region extractors are unsupervised and usually rely on the following algorithms: tree matching, string matching, and clustering. This makes the majority of proposals scalable since they do not rely on a user to provide samples of the regions to be extracted.

3) The efficiency and effectiveness is by far the dimension in which there are more missing data in the literature; furthermore, the available results are not comparable side by side. It is an essential requirement to count on an up-to-date and maintained dataset repository to perform homogeneous and fair empirical evaluations.

4) Region extraction is not an easy task. The proposals in the literature have strong features and drawbacks, but none of them is universally applicable, which keeps this quite an active research field.

III. OVERVIEW OF PROPOSED WORK

According to the above survey, to write specialized programs, called wrappers is the traditional approach for extracting data from web sources that identify data of interest and map them to some suitable format as XML, CSV or relational tables. We have extended this traditional approach with segmentation and parsing. The tool relies on inherent structural features of HTML document for accomplishing the data extraction. Before performing the extraction process, this tool turns the document into parse tree a representation that reflects its HTML tag hierarchy (DOM structure). Further extraction is done automatically or semi automatically by applying extraction rule to the DOM structure.

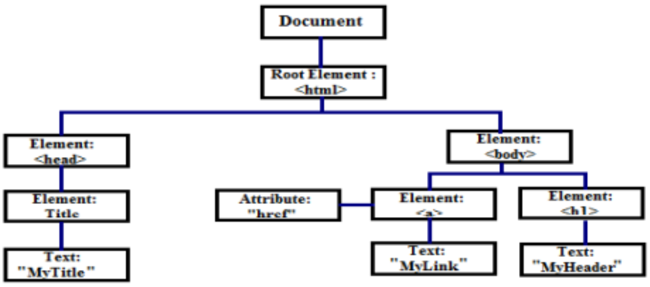

Figure 1. Overview of DOM tree

The Document Object Model most often referred to as DOM is a cross-platform and language independent convention for representing and interacting with objects in HTML. The DOM tree defines the logical structure of documents and the way a document is accessed and manipulated. It is constructed based on the organization of HTML structures (tags, elements, attributes). The HTML DOM views a HTML document as a tree-structure (node-tree). Every node can be accessed through the tree. Their contents can be modified or deleted. New elements can also be created. In this paper, the basic approach of web data extraction process is implemented through the Document Object Model (DOM) tree. Using a DOM tree is an effective way to identify a list or extract data from the web page. Anything found in an HTML document can be accessed, changed, deleted or added using the DOM tree. Fig 1. shows an Overview of the DOM Tree depicting the set of nodes that are connected to one another. The tree starts at the root node and branches out to the text nodes at the lowest level of the tree.

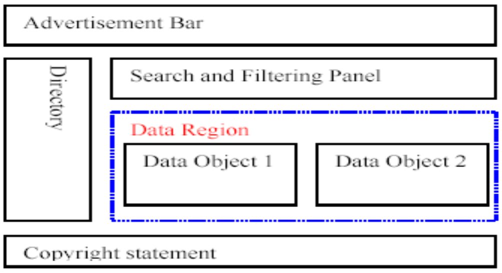

Fig. 2 A schematic view of a webpage

Fig.2 shows the schematic view of a web page on which the harvesting is to be applied. The harvesting process is carried out through the following steps:

1. Creation of config file.

This file acts as a metadata file because it contains the information about the information which has to be extracted from the page.

2. Segmentation of page.

Initially parent element of all child elements (these are the elements which are holding the interested information.) is identified for the purpose of segmentation. Depending upon this parent element and its properties page is segmented into no of segments where there is a high potential of finding the desired information.

3. Parsing segments

These segments are parsed in order to get information. Since there is high possibility that these might not be in the proper format so the decision making has to be there to get the correct information.

Steps involved in decision making.

1. Tag type of the element is checked.
2. Class attribute value is matched.
3. No of children of the given element.
4. Tag type of each child element.

Fig:3 shows the probable design of the proposed harvesting tool

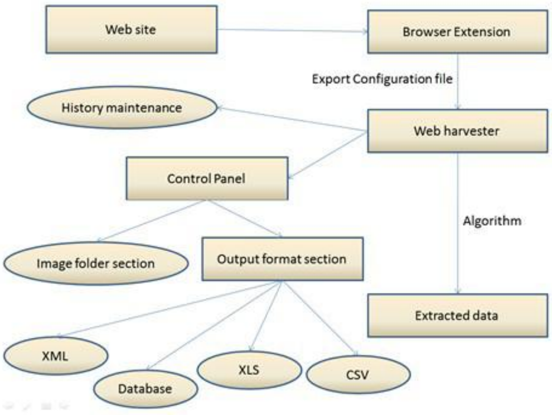

Fig3. Proposed Model

Data Extraction and Mining

1. User will guide the tool for information to be selected for mining.
2. Then the tool would take that configuration file and it will mine data from web pages.

IV. CONCLUSION

The Web Data Mining is the popular technology in the world. There are many research works going on in this field. In this study we have done survey on various applications where web data mining techniques are used. We have analyzed problems in each application study of web data mining. It can be seen that, in the present state of the art web data extraction, there is an inherent tradeoff between the degree of automation and the degree of flexibility of current data extraction techniques. The techniques considering HTML structure uses a number of heuristics to infer a possible schema from HTML tag hierarchy of a target page. Thus some hypothesis on the use of HTML constructs to structure data must be assumed. Implementation of data mining techniques is an important task in any domain. We have extended the study to design web data mining tool on specified problem domain.

REFERENCES

[1]. Jun Kong, Omer Barkol, et al., “Web Interface Interpretation Using Graph Grammars”, IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 42, no. 4, july 2012

[2]. Mohammed Kayed and Chia-Hui Chang, “ FiVaTech: Page-Level Web Data Extraction

[3]. from Template Pages”, IEEE transactions on knowledge and data engineering, vol. 22, no. 2, february 2010

[4]. Jer Lang Hong, “Data Extraction for Deep Web Using WordNet”, IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 41, no. 6, november 2011

[5]. Weifeng Su, Jiying Wang, Frederick H. Lochovsky , “Combining Tag and Value Similarity for Data Extraction and Alignment” IEEE transactions on knowledge and data engineering, vol. 24, no. 7, july 2012

[6]. Zaiqing Nie, Ji-Rong Wen, and Wei-Ying Ma, “Statistical Entity Extraction From Web”

[7]. Luis Tari, Phan Huy Tu, Jo¨ rg Hakenberg, Yi Chen, Tran Cao Son, Graciela Gonzalez, and Chitta Baral “Incremental Information Extraction Using Relational Databases”, IEEE transactions on knowledge and data engineering, vol. 24, no. 1, january 2012

[8]. Hassan A. Sleiman and Rafael Corchuelo, “A Survey on Region Extractors From Web Documents”, IEEE transactions on knowledge and data engineering

[9]. ERCIM NEWS 34 89 April 2012 “Special theme:Big Data”

[10]. A Comparison of Leading Data Mining Tools (ARTICAL) John F. Elder IV & Dean W. Abbott Elder Research