A comparative study on different approaches of data extraction from web

M Lavanya
Sr. Lecturer,
SVEC,
Tirupati, India.
lavanya4_79@rediffmail.com

Dr. M Usha Rani
Associate Professor, Dept. of C S,
SPMVV,
Tirupati, India.
musha_rohan@yahoo.com

Abstract— **This paper probes various methods for extracting data from various HTML sites, web page that group several structured records, where we are going to look into the use of automatically generated wrappers. The huge volume of information on the web is residing in document databases and is indexed by general-purpose search engines where that information is vigorously generated through querying databases — which are referred to as unseen Web databases. In this paper, we have studied a mixture of methods to extract data items from the web pages automatically. Retrieving structured data from deep Web pages is a main problem due to the essential convoluted structures of web pages. A huge number of techniques have been reviewed to address this problem, but all of them have innate margins on that the Web-page-programming-language reliant. A Vision-based approach that is Web-page programming-language-independent. Here we look-into the visual features on the deep Web pages to implement deep Web data extraction, including data record extraction and data item extraction.**

Keywords- language dependent, language independent, web data extraction, Wrappers, DOM Trees

I. INTRODUCTION

On the internet we need lots of information available in HTML format increases rapidly. In the internet URL available we have contemporary bequest systems, as a huge amount of data can be easily entrée and operated. The main cause is that all the web data sources are required to search by users. Using XML we can also overcome some of the hurdles arise in the HTML. So it is difficult to extract data from web pages which ruins a typical and appropriate task. Software modules performed on HTML using wrappers for data extraction.

As per Ricardo A. Baeza-YatesIn [8], a string matching method whose results are not accurate. Most of the systems make that the accurate information of a data record is available in a bordering fragment of the HTML code. The problem with such web pages is that narration of one object may interlace with the narration of some other objects other objects. In this paper we studied an approach to resolve the problem by taking a page, the technique first fragments the page to identify each data record without extracting its data items. Yangon Zhao and Bing Liu developed MDR [3] for this purpose. Specifically, the method also uses visual cues to find data records. Visual information helps the system in two ways:(i) It enables the system to identify gaps that separate data records, which helps to segment data records correctly because the gap within a data record (if any) is typically smaller than that in between data records. (ii) We re-examine system that identifies data records by analyzing HTML tag trees or DOM trees [7]. [4]In response to the user queries, we can generate a document list which is dynamically generated from the hidden web databases. The number of terms extracted from the web always available in templates which is used for description or navigation purposes. The list of stop words to eliminate unwanted terms [1]. Various string matching methods are take up by [6] from web pages to avail information. Here only the textual content differences and similarities. According to Y.L. Hedley, M. Younas, A. James and M. Sanderson an approach that identifies the portions of Web documents that are relevant to a user query by analysing textual contents and their adjacent tag structures. The results of this technique provide efficient method to extract query-related data from Hidden web databases [4].

P. S. Hiremath, Siddu P. Algur presented various approaches where the structured data objects on the web are normally database records retrieved from underlying web databases and display in web pages with some fixed templates[5]. Automatic methods aim to find patterns/grammars from the web pages and then use them to extract data. Examples of automatic systems are IEPAD [5], ROADRUNNER [2], MDR [5], DEPTA [3] and VIPS [11]. These systems make use of the Patricia (PAT) tree for discovering the record boundaries automatically and a pattern based extraction rule to extract the web data. This method has a poor performance due to the various limitations of the PAT tree. ROADRUNNER [6] extracts a template by analyzing a pair of web pages of the same class at a time. The objective of this paper is to extract the data items from a given webpage using visual clues and list based approach. The studied method is implemented in two ways to solve the problem: i) For a web page, the first method (or step) identifies and extracts the data region based on the visual clue (location of data region / data records / data items / on the screen at which the tags are rendered) information of web pages. The algorithm is called **VSAP** (**V**isual **S**tructure based **A**nalysis of web **P**ages [16]). ii) For a relevant data region (i.e. the output of VSAP algorithm),

the second method (or step) identifies the data records and extract the data items from it. The algorithm is here after called **EDIP** (**E**xtraction of **D**ata **I**tems from Web**P**ages) which is a list based approach [17]. It finds the data items formed by all types of tags. We have also reviewed a four-step strategy. a). Find visual representation and convert into a Visual Block tree from a deep web page. b). Retrieve data records from the Visual Block Tree c). Divide extracted data records into data items and arrange data items of the same semantic together. d).Finally, generate Visual Wrapper for the web database on a deep web page for both data record extraction and data item extraction. To our best knowledge, although there are already some works [10], [11], that pay attention to the visual information on Web pages, we mainly study how to perform deep Web data extraction using primarily visual features. The approach is independent of any specific Web page programming language. Although re-examined implementation uses the VIPS algorithm [8] to obtain a deep Web page's Visual Block tree and VIPS needs to analyze the HTML source code of the page, the solution is independent of any specific method used to obtain the Visual Block tree in the sense that any tool that can segment the Web pages into a tree structure based on the visual information, not HTML source code, can be used to replace VIPS in the implementation of ViDE. In this paper, we also studied a new measure, revision, to evaluate the performance of Web data extraction tools. It is the percentage of the Web databases whose data records or data items cannot be perfectly extracted (i.e., at least one of the precision and recall is not 100 percent). For these Web databases, manual revision of the extraction rules is needed to achieve perfect extraction.

II. VARIOUS METHODS FOR DATA EXTRACTION

A. String searching method

The important component of many problems is String searching, where it adds text editing, data retrieval and symbol manipulations [1]. Finding all occurrences of a pattern in a text is the problem in string searching and string matching problem. We have studied interesting in reporting all the occurrences. It is well known that to search for a pattern of length rp in a text of length p (where p > m) the search time is *O (p) in* the worst case (for fixed rp). Moreover, in the worst case at least p - rrt + 1 character must be inspected [1].

B. Road Runner

We have surveyed various techniques for extracting data from HTML sites through the use of automatically generated wrappers [2]. To automate the wrapper generation and the data extraction process, where we have studied an approach to contrast HTML pages and generate a wrapper based on their similarities and differences. We studied that the page-generation process as the result of two separated activities as one is to execute number of queries on the underlying database to generate a source dataset, i.e. a set of tuples of a possibly nested type that will be published in the site pages; b) The serialization of the source dataset into HTML code to produce the actual pages, possibly introducing URLs links, and other material like banners or images.

a) *The Matching approach:* We studied algorithm match which is based on the matching approach for Align, Collapse under Mismatch, and Extract to avoid errors and missing tags in the sources where we studied that compiling HTML code making use of XHTML specification, a avoiding variant of HTML in which tags are required to be properly closed and nested. We have learnt that sources have been pre-processed by a lexical analyzer to transform them into lists of tokens; each token is either an HTML tag or a string value. The matching algorithm works on two physical entities at same time: a list of tokens, called the sample, and a wrapper, i.e., one union-free regular expression. A mismatch helps us to find important information about the wrappers. If mismatch is found then resolve mismatch by generalizing the wrapper. The algorithm succeeds if a common wrapper can be generated by solving all mismatches encountered during the parsing.

Mismatches: Mismatches are categorized into two classes

- String mismatches: When different strings occur in corresponding positions of the wrapper and sample then mismatch occurs.
- Tag mismatches : mismatches between different tags on the wrapper and the sample, or between one tag and one string.

C. MDR

MDR algorithm, where we have made two interpretations about data records in a Web page and an edit distance string matching algorithm [2] to find data records which was developed by Yanhong Zhai and Bing Liu. The interpretations are: A group of data records that contains descriptions of a set of similar objects are typically presented in a contiguous region of a page and are formatted using similar HTML tags. Such a region is called a *data record region* (or *data region* in short). The problem with this approach is that the computation is prohibitive because a data record can start from any tag and end at any tag. A set of data records typically does not have the same length in terms of its tag strings because it may not contain exactly the same pieces of information. The next observation helps to deal with this problem. The nested structure of HTML tags in a Web page naturally forms a *tag tree*.

Advantages:

a) Identify the data records based on the keyword search

b) It identifies the relevant data region containing the search results but also extracts from all other sections of the page.

c) Comparison of generalized nodes is based on string comparison using normalized edit distance method

Disadvantages:

a) Due to noisy information, it may find wrong combination of sub trees which leads to inefficiency.

b) MDR [10] basically exploits the regularities in the HTML tag structure directly. It is often very difficult to derive accurate wrappers entirely based on HTML tags. MDR algorithm makes use of the HTML tag tree of the web page to extract data records from the page.

c) An incorrect tag tree may be constructed due to the misuse of HTML tags, which in turn makes it impossible to extract data records correctly.

Segmenting the Web page to identify individual data records. It does not align or extract data items in the data records. Since this step is an improvement to our previous technique MDR [21], below we give a brief overview of the MDR algorithm and present the enhancements made to MDR in this work. We also call the enhanced algorithm MDR-2 (version 2 of MDR).Given a Web page, the algorithm works in three steps:

Step 1: Building a HTML tag tree of the page. In the new system, visual (rendering) information is used to build the tag tree. Step 2: Mining data regions in the page using the tag tree. A *data region* is an area in the page that contains a list of similar data records. Instead of mining data records directly, which is hard, MDR mines data regions first and then finds data records within them.

Step 3: Identifying data records from each data region. The main enhancement to the MDR algorithm is the use of visual information to help building more robust trees and also to find more accurate data regions.

D. DEPTA (Data Extraction based on Partial Tree Alignment):

We now re-examine the partial tree alignment method for data extraction [3]. The key task is how to match corresponding data items or fields from all data records. There are two sub-steps:

i). Produce one rooted tag tree for each data record: After all data records are identified, the sub-trees of each data record are rearranged into a single tree. Each data record may be contained in more than one sub-tree of the original tag tree of the page, and each data record may not be contiguous. Thus, this sub-step is needed to compose a single tree for each data record (an artificial root node may also need to be added).

ii). Partial tree alignment: The tag trees of all data records in each data region are aligned using our partial alignment method which is based on tree matching. It should be noted that in the matching process, we only use tags. No data item is involved. We simply choose the possible sub-tree alignment that appears the earliest in the tree. Thus, it is not designed for complicated decision strategy.

Advantages:

- Uses visual information to find the data records.
- Visual information is utilized to infer the structural relationship among tags and to construct a tag tree.

Disadvantages:

- Computation time for constructing the tag tree is overhead.
- It fails to identify some of the data records.
- The tag tree can be built correctly only as long as the browser is able to render the page correctly.

DEPTA [15] uses visual information (locations on the screen at which the tags are rendered) to find data records. Rather than analyzing the HTML code, the visual information is utilized to infer the structural relationship among tags and to construct a tag tree. But this method of constructing a tag tree has the limitation that, the tag tree can be built correctly only as long as the browser is able to render the page correctly. The computation time for constructing the tag tree is also an overhead.

E. Query-related data extraction

According to Y.L. Hedley, M. Younas, A. James and M. Sanderson, the extraction of data relevant to a user query from a Hidden Web document - which we refer to as *queryrelated data*. In the query-related data extraction approach there are three phases.

a) To Represent and Process the Document Contents.

The documents retrieved from databases into a list of tag segments and text segments. Tag segments contain starting tags, ending tags or single tags. Text segments are texts that exist in between tag segments. Each text segment can further be identified by its adjacent tag segments. Adjacent tag segments of a text segment are defined as the tags that are located before and after the text segment. A text segment is then defined as follows. *TextSegment = {text, tag, tag}*

b) Recognizing Web page template

Hidden Web databases are often presented using one or more templates, the documents are extracted. The mechanism to detect templates are: (i) Text segments of documents are analysed based on textual contents and their adjacent tag segments (ii) An initial template is identified by examining the first two sample documents (iii) The template is then generated if matched text segments along with their adjacent tag segments are found from both documents (iv) Subsequent documents retrieved are compared with the template generated. Text segments that are not found in the template are extracted for each document to be further processed (v) when no matches are found from the existing template, document contents are extracted for the generation of future templates.

c) Determine the Similarities between Text Segments of Different Documents

The text segments extracted from documents are further analysed. This process identifies document contents that have not been found in the templates generated from the initial sample documents. In this phase, the text of a segment is represented as a vector of terms with weights. A term weight is

obtained from the frequencies of the term that appears in the segment. Cosine similarity [7] is computed for the text segments of different documents that are generated from the same template in order to determine their similarities. The computation of similarity for two text segments is given as follows. Where TERM is the weight of term k in the first text and TERM is its weight in the second text. The similarities are computed for text segments with identical adjacent tag segments only. Two segments are considered to be similar if the similarity exceeds a threshold value. Such a threshold value is determined experimentally. This process extracts text segments that are significantly different from textual content and tag structures contained in templates.

F. VSAP (Visual Structure based Analysis of web pages[16])

It finds the data regions formed by all types of tags using visual clues. Given a relevant data region (i.e. the output of VSAP algorithm), the second method (or step) identifies the data records and extract the data items from it. The algorithm is here after called EDIP (Extraction of Data Items from Web Pages) which is a list based approach [17]. It finds the data items formed by all types of tags. This technique has two process to be involved as one is to identifying individual data records in a page. And the other is to align and extract data items from the identified data records. Specifically, this method also uses visual cues to find data records.

G. ViDE

We investigate the visual regularity of the data records and data items on deep Web pages, [6] Vision-based Data Extractor (ViDE), to extract structured results from deep Web pages automatically. ViDE is primarily based on the visual features human users can capture on the deep Web pages while also utilizing some simple non visual information such as data types and frequent symbols to make the solution more robust. ViDE consists of two main components, Vision based Data Record extractor (ViDRE) and Vision-based Data Item extractor (ViDIE). By using visual features for data extraction, ViDE avoids the limitations of those solutions that need to analyze complex Web page source files. Similarly for the GDS dataset we have comparative analysis of the parameter precision, recall and revision of various approaches.

In ViDE[6] , we have studied the method as the following scenario- that a given a sample deep Web page from a Web database, acquire its visual symbol and transform it into a Visual Block tree from which data records are extracted. Then partition the extracted data records into data items and align the data items of the same semantic together then spawn visual wrappers for the Web database, based on sample deep Web pages such that both data record extraction and data item extraction for new deep Web pages that are from the same Web database can be carried out more efficiently. Instead of extracting data records from the deep Webpage directly, we first extract the data region, and then, extract data records from the data region. PF1 and PF2 indicate that the data records are the primary content on the deep Web pages and the data region is centrally located on these pages. Data record extraction is to discover the boundary of data records based on the LF and AF features. That is, we attempt to determine which blocks belong to the same data record. We achieve this in the following three phases: 1. Phase 1: Filter out some noise blocks. 2. Phase 2: Cluster the remaining blocks by computing their appearance similarity. Phase 3: Discover data record boundary by regrouping blocks.

III. COMPARATIVE ANALYSIS

Thus we have come across several methods [2], [4], [5], [6], [14] for structured data extraction, which is also called wrapper generation. The first method [2] is to manually written an extraction program for each web site based on observed format patterns of the site. This manual approach is very labour intensive and time consuming. Hence, it does not scale to a large number of sites. The second approach [4] is wrapper induction or wrapper learning, which is currently the main technique. Wrapper learning works as follows: The user first manually labels a set of trained pages. A learning system then generates rules from the training pages. The resulting rules are then applied to extract target items from web pages. These methods either require prior syntactic knowledge or substantial manual efforts. The third approach [5] is the automatic approach. The structured data objects on a web are normally database records retrieved from underlying web databases and displayed in web pages with some fixed templates. Automatic methods aim to find patterns/grammars from the web pages and then use them to extract data. Examples of automatic systems are ROADRUNNER [6], MDR [10], DEPTA [15] and VIPS [11].The main hurdle of all the above said methods, make use of the Patricia (PAT) tree for discovering the record boundaries automatically and a pattern based extraction rule to extract the web data. The main disadvantage of this method has a poor performance due to the various limitations of the PAT tree. ROADRUNNER [6] extracts a template by analyzing a pair of web pages of the same class at a time. It uses one page to derive an initial template and then tries to match the second page with the template.

However, this method also fails to identify some of the data records. VIPS [9] is based on certain visual cues within the content of a node of a parse tree. Yu et al. [13] suggest that tags such as <HR>, which is used to create a horizontal line, as well as attributes that indicates a change in background colour can indicate the beginning or end of a segment. These segmentation cues are seen as "visual separators" and are classified as horizontal or vertical lines.

VIPS Advantages:

a) The content structure of the page is created by merging visual blocks that are not divided by separators.

b) Knowledge on document structure is used to enhance the quality.

TABLE 1: CATEGORIZED FEATURE OF DATA EXTRACTION APPROACHES

MDR	VSAP	DEPTA
MDR identifies the data records based on keyword search	VSAP is purely dependent on the visual structure of the web page only	comparison of sub-trees is made by simple tree matching algorithm
It make use of any text or content mining	It does not make use of any text or content mining	comparison is purely numeric
Overhead of performing keyword search on the web page.	It overcomes overhead of performing keyword search on the web page.	It scales well with all the web pages
only identifies the relevant data region containing the search result records	also extracts records from all the other sections of the page	Extracts visual information to segment data records, which is more accurate
It may find wrong combinations of sub-trees	visual gaps between data records help to deal with noisy problem	comparison of sub-trees is made by simple tree matching algorithm
time complexity of the order O (NK) without considering string comparison where N is the total number of nodes in the tag tree and K is the maximum number of tag nodes that a generalized node	complexity of the order of O(n), where n is the number of tag comparisons made	the time complexity of the order O (k2) without considering tree matching, where k is the number of trees.

VIPS Disadvantages:

a) It is dependent on number of heuristic rules which cannot be applied for most of the web pages.

b) It doesn't correctly identify the data regions.

In TABEL 1, we have compared various approaches for data extraction from the web like MDR, VASP and DEPTA.

We also analyzed the comparison of two state-of-the-art existing systems, DEPTA [15] (which is an improvement of [10]) and MDR [10]. We do not compare it with the method in [5] and the method in [14] here as it is shown in [10] that MDR is already more effective than them. In the studied VSAP system, the data region identification is independent of specific tags and forms. Unlike [10], where an incorrect tag tree may be constructed due to the misuse of HTML tags, there is no such possibility of erroneous tag tree construction in case of VSAP, because the hierarchy of tags is constructed based on the visual cues on the web page. In case of [14] and [15], the entire tag tree needs to be scanned in order to mine data regions, but VSAP doesn't scan the entire tag tree, but it only scans the largest child of the <BODY> tag. We analyzed the experiments on a larger sample of concealed Web documents. The dilemma is that existing approaches are either erroneous or make many strong *assumptions.* An approach based on a combination of VSAP and EDIP is examined to dig for the data region and data items in a web page automatically. ViDE approach is also used to extract vision based data extraction.

Table II: PERFORMANCE ANALYSIS FOR TWO DIFFERENT DATA SETS SDS AND GDS SETS

	SDS Data set			GDS Data set		
Various Approaches For data extraction	Precision	Recall	Revision	Precision	Recall	Revision
AEiEDWD	67.4	54.8	38.6%	76.2	72.4	33.4
ViDIE	95.6	98.4	11.6	96.3	97.2	14.1
DEPTA	66.1	54.1	37.6	75.3	71.6	32.8
ViDRE	98.5	97.8	10.9	98.7	97.2	12.4
MDR	78.7	47.3	63.8	85.3	53.2	55.2

Performance analysis of five different approaches of data extraction with respect to parameters of precision, revision and recall to assess the performance of different approaches are shown in Table 2.

In Figure 1 we have analysed that ViDRE has 2.4% more precision to ViDIE, 13.4% less precision to MDR, 22.5% less precision to AEiEDWD and 23.4% less precision to DEPTA to assess the performance for GDS data set. Like-wise the revision and recall can be analysed for GDS dataset.

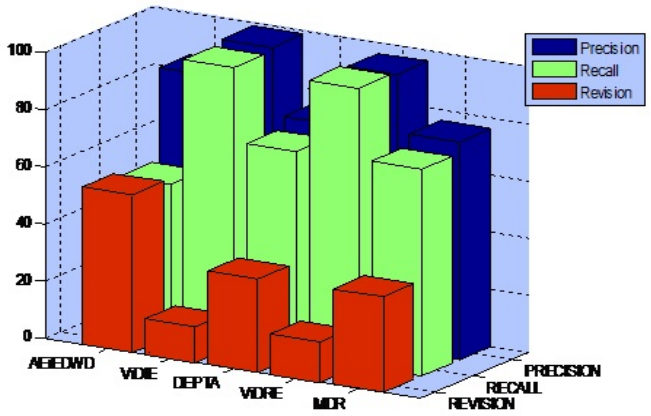

Figure 1: Performance analysis of Dataset GDS

Similarly, In Figure 2 we have analysed that ViDRE has 2.9% more precision to ViDIE, 19.8% less precision to MDR, 31.1% less precision to AEiEDWD and 32.4% less precision to DEPTA to assess the performance for SDS data set. Like-wise the revision and recall can be analysed for SDS dataset.

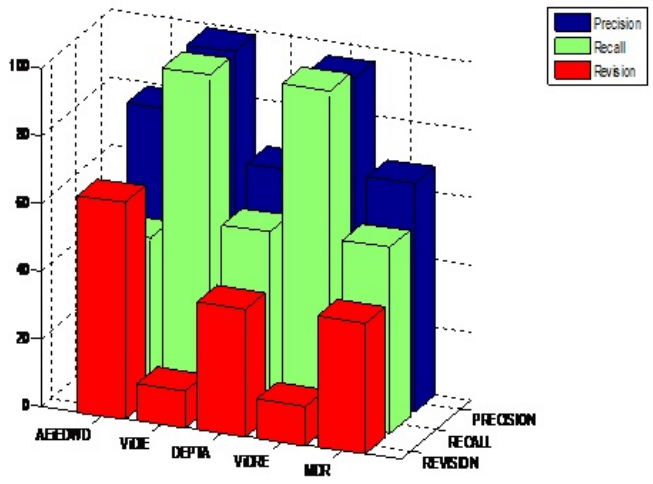

Figure 2: Performance analysis of Dataset SDS

IV. CONCLUSION

We can search a string in worst case time proportional to its length for textual information available in number of files. This is achieved by using a Patricia tree [Mor68] as index. The solution from string matching method needs *O(n)* extra space and pre-processing time, where n is the size of the text [Wei73, McC76, Mor68, MRS0, KBG87]. An important aspect re-examined comment is related to the quality of the data extracted by the wrappers. To extract structured data from Web pages. To extract query-related data in order to obtain terms and frequencies with a higher degree of accuracy. The techniques examine text segments along with their adjacent tag structures rather than analysing document contents in a treelike structure as in [2]. This provides an effective mechanism for template detection. In this paper, we have reviewed how to extract structured data from web pages. Although the problem has been studied by several researchers, existing techniques are either inaccurate or make many strong assumptions. ViDE approach consists of four primary steps: Visual Block tree building, data record extraction, data item extraction, and visual wrapper generation. Highly accurate experimental results provide strong evidence that rich visual features on deep Web pages can be used as the basis to design highly effective data extraction algorithms.

REFERENCES

[1] Ricardo A. Baeza-Yates, "Algorithms for String Searching: A Survey" Data Structuring Group Department of Computer Science University of Waterloo Waterloo, Ontario, Canada N2L 3G1.

[2] Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo, RoadRunner: Towards Automatic Data Extraction from Large Web Sites", Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

[3] Yanhong Zhai, Bing Liu, "Web Data Extraction Based on Partial Tree Alignment" International World Wide Web Conference Committee (IW3C2), *WWW 2005*, May 10-14, 2005, Chiba, Japan.ACM 1-59593-046-9/05/0005.

[4] Y.L. Hedley, M. Younas, A. James, M. Sanderson,"Query-Related Data Extraction of Hidden Web Documents**",** *SIGIR'04*, July 25–29, 2004, Sheffield, South Yorkshire, UK.ACM 1-58113-881-4/04/0007.

[5] P. S. Hiremath, Siddu P. Algur ,"Extraction of Data from Web Pages: A Vision Based Approach", World Academy of Science, Engineering and Technology 51 2009.

[6] Wei Liu, Xiaofeng Meng, Weiyi Meng, "ViDE: A Vision-Based Approach for Deep Web Data Extraction", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. X, XXXXXXX 2010.

[7] J. Hammer, J. McHugh, and H. Garcia-Molina, "Semi-structured Data: The TSIMMIS Experience," Proc. East-European Workshop Advances in Databases and Information Systems (ADBIS), pp. 1-8, 1997.

[8] Gusfield, D. *Algorithms on strings, tree, and sequence*, Cambridge. 1997.

[9] Chen, W. New algorithm for ordered tree-to-tree correction problem. *Journal of Algorithms*, 40:135–158, 2001.

[10] D. Embley, Y. Jiang, and Y. K. Ng. Record-boundary discovery in Web documents. *ACM SIGMOD Conference, 1999.*

[11] Kushmerick, N. Wrapper Induction: Efficiency and Expressiveness *Artificial Intelligence*, 118:15-68, 2000. Clustering-based Approach to Integrating Source Query].

[12] G.O. Arocena and A.O. Mendelzon, "WebOQL: Restructuring Documents, Databases, and Webs," Proc. Int'l Conf. Data Eng. (ICDE), pp. 24-33, 1998.

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

[14] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma, "Block-Level Link Analysis," Proc. SIGIR, pp. 440-447, 2004.

[15] D. Cai, S. Yu, J. Wen, and W. Ma, "Extracting Content Structure for Web Pages Based on Visual Representation," Proc. Asia Pacific Web Conf. (APWeb), pp. 406-417, 2003.

[16] C.-H. Chang, M. Kayed, M.R. Girgis, and K.F. Shaalan, "A Survey of Web Information Extraction Systems," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 10, pp. 1411-1428, Oct. 2006.