Annotation for Query Result Records based on Domain-Specific Ontology

S. Lakshmana Pandian[1] and R.Punitha[1]

[1]Department of Computer Science & Engineering
Pondicherry Engineering College
Puducherry, India

ABSTRACT

The World Wide Web is enriched with a large collection of data, scattered in deep web databases and web pages in unstructured or semi structured formats. Recently evolving customer friendly web applications need special data extraction mechanisms to draw out the required data from these deep web, according to the end user query and populate to the output page dynamically at the fastest rate. In existing research areas web data extraction methods are based on the supervised learning (wrapper induction) methods. In the past few years researchers depicted on the automatic web data extraction methods based on similarity measures. Among automatic data extraction methods our existing Combining Tag and Value similarity method, lags to identify an attribute in the query result table. A novel approach for data extracting and label assignment called Annotation for Query Result Records based on domain specific ontology. First, an ontology domain is to be constructed using information from query interface and query result pages obtained from the web. Next, using this domain ontology, a meaning label is assigned automatically to each column of the extracted query result records.

Keywords: Data Extraction, Deep Web, Wrapper Induction, Data Record Alignment, Domain Ontology.

1. INTRODUCTION

Deep Web primarily contains some dynamic pages returned by the underlying databases. We call the accessed online databases on the Web as “Web Databases (WDB)". WDB responds to a user query to the relevant data, either structured or semi structured, embedded in HTML pages (called query result pages in this paper). To utilize this data, it is necessary to extract it from the query result pages. By analysing and summarizing web data, we can find latest market trends, price details, product specification etc. Manual data extraction is time consuming and error prone.

Hence, Automatic data extraction plays an important role in processing results provided by search engines after submitting the query by the user. The wrapper is an automated tool which extracts Query Result Records (QRRs) from HTML pages returned by search engines. A query result page returned from a WDB has multiple QRRs. Each QRR contains multiple data values each of which describes one aspect of a real-world entity. There is a high demand for collecting data of interest from multiple WDBs.

For example, once a book comparison shopping system collects multiple result records from different book sites, it needs to determine whether any two QRRs refer to the same book. A book can be compared with different websites by ISBN number. Suppose, the ISBNs are not available, then their titles and authors could be compared. The system also needs to list the prices offered by each site. Thus, the system needs to know the semantics of each data value. Unfortunately, the semantic labels of data values are often not provided in result pages. Having semantic labels for

data value is not only important for the above record linkage task, but also for storing collected QRRs into a database table for later analysis.

Early applications require tremendous human efforts to annotate data units manually, which severely limit their scalability. Presently, Semantic annotation bases on correct extraction of query results. Now automatic web data extraction has been relatively matured. In this work, the method of automatic web data extraction defined in Combining Tag and Value Similarity (CTVS) for data extraction and alignment [1] is adapted. Precisely, we propose to integrate the label assignment for the attribute in the extracted QRRs based on the construction of an ontology domain to give a better performance results.

A Web database responds to a user query to the relevant data in the form of query result pages. The query result pages may be in a structured or semi-structured format. It is necessary to extract the data values from the query result pages for the efficient utilization of data. Now-a- days, automatic data extraction approach is popular for dynamically extracting the data from the web database.

The goal of automatic data extraction method is to extract the relevant data from the query result pages. Thus, the data values returned from the underlying database are usually encoded in the result pages for human browsing. Also, the returned data values must be understandable and processable by the machines. Because, it is essential for many applications such as deep web data collection and comparison shopping, these extracts the data values with meaningful labels. However, due to the automatic nature of the CTVS [1] approach, the data extracted will be anonymous names. The annotated method is necessary for making meaningful names to the extracted data values.

The current annotation methods are annotated from the single website for the query result pages. It does not give better performance when the label values are not present in a website. The proposed method annotation for query result records based on domain specific ontology is used to construct an ontology domain from different websites. It would give better performance when integrating with CTVS data extraction method.

The following is the description of how the chapters are getting organized. In this section, we discuss about various approaches of the data extraction methods. Section 2 describes the related work of existing web data extraction methods. The Section 3 discusses in detail about the existing web data extraction method, combined tag and value similarity. Section 4 deals the proposed mechanism for enhancing the automatic data extraction method. Section 5 explains the implementation of the existing system and experimental results obtained. Section 6 concludes by stating the work

2. RELATED WORK

RoadRunner [2] – This method is used for extracting the data from the web page without any human intervention. It starts with any page as its initial page template and then compares this template with each new page. If the template cannot generate the new page, it is fine-tuned. However, RoadRunner suffers from several limitations. (1) When RoadRunner finds that the current template cannot generate a new page, it searches through an exponential size page schema trying to fine tune the template. (2) RoadRunner assumes that the template generates all HTML tags, which does not hold for many Web databases.(3) RoadRunner assumes that there are no disjunctive attributes, which the authors of RoadRunner admit does not hold for many query result pages. (4) Data labels/annotation is not addressed in RoadRunner, since it mainly focuses on the data extraction problem.

Omini [3] – This method uses several heuristics to extract a subtree that contains data strings of interest from an HTML page. Then, another set of heuristics is used to find a separator to segment the minimum data object rich subtree into data records. However, Omini has a low effectiveness for extracting data records than other extraction methods.

ExAlg [4] – This method is based on template extraction from the query result page. It performs template extraction in two stages. The first stage is Equivalence Class Generation Stage (ECGM) and second is analysis stage. ECGM stage computes equivalence classes. i.e, set of tokens having the same frequency of occurrence in every page. This is performed by a FindEquiv sub module. There may be many equivalence classes. But ExAlg only considers equivalence classes that are large and contain tokens which occur in a large number of pages. Such equivalence classes are called Large and Frequently Occurring Equivalence Classes (LFEQs). It is very unlikely for LFEQs to be formed by chance. LFEQs are formed by tokens associated with the same type constructor in the template.

IePAD [5] – This method is one of the first IE systems that generalize extraction patterns from unlabelled Web pages. This method exploits the fact that if a Web page contains multiple (homogeneous) data records to be extracted, they are often rendered regularly using the same template for better visualization. Thus, repetitive patterns can be discovered if the page is well encoded. Therefore, learning wrappers can be solved by discovering repetitive patterns. IePAD uses a data structure called PAT trees which is a binary suffix tree to discover repetitive patterns on a Web page. Since such a data structure only records the exact match for suffixes, IePAD further applies a center star algorithm to align multiple strings which start from each occurrence of a repeat and end before the start of next occurrence.

DeLa [6] – This method is an extension of IePAD. The method [7] removes the interaction of users in extraction rule generalization and deals with nested object extraction. The wrapper generation process in DeLa works on two consecutive steps. First, a Data-rich Section Extraction algorithm (DSE) is designed to extract data-rich sections of the Web pages by comparing the DOM trees for two Web pages (from the same Web site), and discarding nodes with identical sub-trees. Second, a pattern extractor is used to discover continuously repeated (C-repeated) patterns using suffix trees. By retaining the last occurrence for each discovered pattern, it discovers new repeated patterns from the new sequence iteratively, forming nested structure. Since a discovered pattern may cross the boundary of a data object, It tries K pages and selects the one with the largest page support. Again, each occurrence of the regular expression represents one data object. The data objects are then transformed into a relational table where multiple values of one attribute are distributed in multiple rows of the table. Finally, labels are assigned to the columns of the data table by four heuristics, including element labels in the search form or tables of the page and maximal-prefix and maximal-suffix shared by all cells of the column. Multiple patterns (rules) and it is hard to decide which is correct.

TISP [8] – This method constructs wrappers by looking for commonalities and variations in sibling tables in sibling pages (i.e., pages that are from the same website and have similar structures). Commonalities in sibling tables represent labels, while variations represent data values. Matching sibling tables are found using a tree-mapping algorithm applied to the DOM tree representation of tagged tables in sibling pages. Using several predefined table structure templates, which are described using regular expressions, TISP "fits" the tables into one of the templates allowing the table to be interpreted. The TISP is able to handle nested tables as well as variations in table structure (i.e., optional columns) and is able to adjust the predefined templates to account for various combinations of table templates. The whole process is fully automatic and experiments show that TISP achieves an overall F-measure of 94.5% on the experimental data set.

However, TISP can only extract data records embedded in HTML<table> tags and only work when sibling pages and tables are available.

NET [10] – It extract data items from data records even it handles nested data records also. There are two main steps. Building tag tree is difficult because page may contain erroneous and unbalanced tags. This is performed based on nested rectangles. For this four boundaries of each HTML element are determined by calling embedded parsing and rendering engine of a browser. A tree is constructed based on containment check, whether one rectangle contained inside another.

NET algorithm traverses the tag tree in post order (bottom up) in order to find nested data records which are found at lower levels. These tasks are performed using two algorithms traverse and output. Traverse algorithm finds nested records by recursive call if the depth of sub-tree from the current node greater than or equal to three. A simple tree matching algorithm is used here. Data items are aligned after tree matching. All aligned data items are then linked in such a way that a data item will point to its next matching data item. The output algorithm will put a linked data into a relational table. A data structure used is linked list of one dimensional array which represents columns. This data structure makes it easier to insert columns in appropriate location for optional data items. All linked data items are put in the same column. A new row is started if an item is being pointed to by another item in an earlier column.

DePTA [11] – This method proposes the algorithm of partial tree alignment technique for aligning the extracted records. The intuition that the gap within a QRR is typically smaller than that between QRRs is used to segment data records and to identify individual data records. The data strings in the records are then aligned, for those data strings that can be aligned with certainty, based on tree matching. The DOM tree is constructed from the HTML result page based on the tags which covers the data values are returned from the search result page, which is submitted by the user query. This step finds the data region by comparing tag strings associated with individual nodes including descendants and combination of multiple adjacent nodes. Similar nodes are labelled as data region. Generalized node is introduced to denote each similar individual node and node combination. Adjacent generalized nodes form a data region. Gaps between data records are used to eliminate false node combinations. Visual observations about data records states that the gap between the data records in a data region should be no smaller than any gap within a data record. Data records are identified from generalized nodes.

There are two cases in which data records are not in contiguous segment. Next, the Data Item extractor is performed based on partial tree alignment technique to match the corresponding data item or fields from all data records. The two sub-steps are Production of one rooted tag tree for each data record. The subtrees of all data records are arranged into a single tree. Partial tree alignment Tag trees of data records in each data region are aligned using partial alignment. This is based on tree matching. No data item is involved in the matching process. Only tag nodes are used for matching. Tree edit distance between two trees is a cost associated with a minimum set of operations needed to transform A into B. Restricted matching algorithm called simple tree matching is used which will produce maximum matching between two trees.

ViNTs [12] – This method uses both visual and tag features to learn a wrapper from a set of training pages from a website. It first utilizes the visual data value similarity without considering the tag structure to identify data value similarity regularities, denoted as data value similarity lines, and then combines them with the HTML tag structure regularities to generate wrappers. Both visual and non visual features are used to weight the relevance of different extraction rules. For this method, the result page must contain at least four QRRs, and one no-result page is required to build a wrapper. The input to the system is the URL of a search engine's interface

page, which contains an HTML form used to accept user queries. The output of the system is a wrapper for the search engine.

In this method, several result pages, each of which must contain at least four QRRs, and one no-result page is required to build a wrapper. If the data records are distributed over multiple data regions only the major data region is reported. It requires users to collect the training pages from the website including the no-result page, which may not exist for many web databases because they respond with records that are close to the query if no record matches the query exactly. The pre learned wrapper usually fails when the format of the query result page changes. It requires no result page. Hence, it is necessary for ViNTs to monitor format changes to the query result pages, which is a difficult problem. Both visual and non visual features are used to weight the relevance of different extraction rules. The final resulting wrapper is represented by a regular expression of alternative horizontal separator tags (i.e., <HR> or

), which segment descendants into QRRs.

ViPER [13] – This method proposes the visual features for extracting the records from the query result pages. It is a fully automated information extraction tool which works on the web page containing at least two consecutive data records which exhibits some kind of structural and visual similarity. ViPER extracts relevant data with respect to the user's visual perception of the web page. Then Multiple Sequence Alignment (MSA) method is used to align these relevant data regions. Hence this method is not able to find the nested structure data. HTML documents can be viewed as labelled unordered trees. A labelled unordered tree is a directed acyclic graph T = (V, E, R, N) where V is sets of vertices, E is sets of edges, R is rooted and N is the label function N: V X L where L is a string.

CTVS deals the Tag and Value similarity, for automatically extracting 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 in a table where the data values of the same attribute are put into the same column. This method is that mainly it handles two cases, first when QRRs are not contiguous, as the query result page often contains auxiliary information irrelevant to the query such as a comment, recommendation, advertisement, navigational panels or information related to hosting site of the search engine and second nested structure that may exist in the QRRs. Also a new record alignment algorithm is designed which aligns the attributes in a record, first pairwise and then holistically, by combining Tag and Data value similarity information. CTVS (Combined Tag and Value Similarity) consists of following two-steps, to extract the QRRs from a query result page .There are Record extraction and Record alignment. In Record extraction, it identifies the QRRs (Query Result Records) in P and involves the following steps: a) Tag tree construction, b) Data region identification, c) Record segmentation, d) Data region merge, e) Query result section identification.

In Record alignment, it 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. QRR alignment is performed by three-step data alignment method that combines tag and value similarity. a) Pairwise QRR alignment - It aligns the data values in a pair of QRRs to provide the evidence for how the data values should be aligned among all QRRs. b) Holistic alignment - It aligns the data values in all the QRRs. c) Nested structure processing – It identifies the nested structures that exist in the QRRs. Among the above discussed web data extraction methods, CTVS can handle both non contiguous and nested structure data but none of the method label assignment.

In related to label assignment, DeLa [6] is a wrapper tool which automatically extracts the data from the web site and assigns meaningful labels to data. Complex search forms are used by this method rather than using keywords to keep track of pages that querying back end database. DeLa [6] often produces multiple patterns and it is hard to decide which is correct. It does not support

the non continuous query result records. It labels columns with labels only from the query result page or query interface from the same website.

TISP [8] uses the labels that its finds pages that are from the same website and have similar structures, when constructing a wrapper for a web page to annotate the data extraction result. TISP++ [14] further augments TISP by generating OWL ontology for a web site. For each table label, TISP++ generates an OWL class. The label name becomes the class name. If a label pairs with an actual value, TISP++ generates an OWL data type property for the OWL class associated with this label. After the OWL ontology is generated, TISP++ automatically annotates the pages from this Web site with respect to the generated ontology and outputs it as an RDF file suitable for querying using SPARQL. The limitation of TISP++ is that it only generates OWL ontology for a single set of sibling pages from the same Web site and does not combine ontology generated from different Web sites in a domain. If the generated ontology is unable to capture all the data semantics of the site (e.g., when voluntary labels are not available in the Web pages), then when it is applied to annotate the rest of the pages from the same Web site, there may still be some data that cannot be labelled off.

Lu et al. [15] aggregates several annotators, most of which are based on the same heuristics as used in the system [6]. One of the unique proposed annotators, schema value annotator, employs an integrated interface schema and tries to match the data values with attributes in the "global" schema. Once a match is discovered, the attribute name from the global schema can be used to label the matched data values.

3. DATA EXTRACTION AND ALIGNMENT

A novel data extraction and alignment method called combined both tag and value similarity for data extraction and alignment (CTVS [1]). It 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. The architecture of the CTVS is shown in Figure 1.

The novel aspect of this method is that mainly it handles two cases, first when the QRR are not contiguous, as the query result page often contains the irrelevant information such as comments, advertisements, navigational panel or information related to hosting site of search engine and second QRR may have nested structure data.

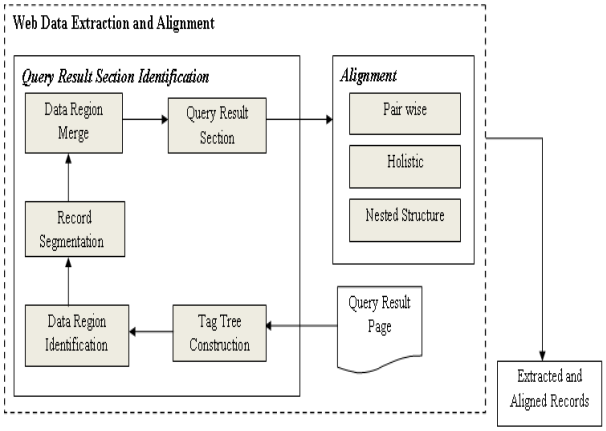

Fig.1. Architecture of CTVS method

In Data Extraction Component, we use the following two-step methods, called Combining Tag and Value Similarity (CTVS), to extract the QRRs from a query result page and align the record using three alignment techniques namely pairwise, holistic and nested structure. CTVS consists of following two-steps, to extract the QRRs from a query result page P.

1. Record extraction, identifies the QRRs (Query Result Records) in P and involves the following steps.

(i) Tag tree construction
(ii) Data region identification
(iii) Record segmentation
(iv) Data region merge
(v) Query result section identification

2. Record alignment, aligns the data values of the QRRs in P into a table so that the data values for the same attribute are aligned into the same table column. QRR alignment is performed by three-step data alignment method that combines tag and value similarity.

(i) Pairwise QRR alignment
(ii) Holistic alignment
(iii) Nested structure processing

4. ONTOLOGY CONSTRUCTION

The goal of the ontology construction component is to build ontology for a domain using the query interfaces and query result pages from Websites within the domain. The ontology construction comprised of four modules. They are query result pages, query interfaces, primary labeling, matching. Each and every module will be described in detail.

Pseudo code for ontology construction:

Step1: Analyse the query interface
Step 2: Analyse the query result page
Step 3: Data wrapping method
Step 4: Primary labeling based on step 1 and step 2
Step 5: Matching
Step 6: Construct domain ontology

1) Query Interface Analysis: Some Web sites support the advance search, shown in Fig.2. Compared with keyword search, detailed query interface has more attributes to allow users to specify more detailed query conditions. Various attributes are in the form of combining a label with a text/selection list component shown below. Some labels in the query interface integrate with attribute in the format of '\Title', '\Format', which is in the web database. In few query result pages, attribute value (along with label) also appear in the result page.

2) Query Result Page Analysis: For a website, the same templates are constructed for displaying the query result page. So, all result pages have the same navigation bar, advertisements and decorative information. The related data displayed in the result page for the given query appear in a specific area. For instance, Fig. 3 shows the part of the query result page specifying the attribute "\Title" in the interface page. We can observe several phenomena as follows: Different data records almost have the same number of attributes, and their structure is also similar to all data

records. In data records, font colour and font size of attribute values belonging to the same concept often share common features.

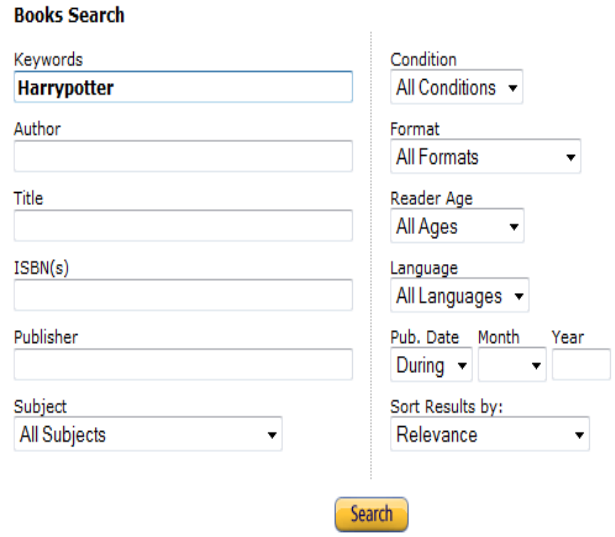

Fig.2: Example of Advanced Query Interface from amazon.com

Fig. 3: Example for query result page from amazon.com

3) Data Wrapping: The web data extracted from the query result pages for a given query by a user is obtained, method called Data Wrapping. Precisely, CTVS shows better performance among all other automatic web data extraction methods. Hence the same method can be used for Data Wrapping.

4) Primary Labeling: To assign labels to the columns of the data table containing the extracted query result record (i.e., to understand the meaning of the data values), the heuristics information proposed in Wang and Lochovsky [2003] is used.

5) Matching: A matcher identifies the matching between query interface and the query result tables from the query result pages from different websites within the same domain.

6) Ontology Construction: Finally, the ontology construction step uses the matching results to construct the ontology.

7) *Annotation:* Assign the label values to the extracted data records based on the constructed domain ontology.

5. EXPERIMENTAL RESULTS

Our annotation method is used to annotate the extracted query result records based on the domain ontology. The CTVS data extraction method is used to extract the query result records from the query result page and align the records in the table format. Then annotate the query result records based on the domain-ontology. This method is classified into major modules as follows: Data Wrapping, Primary Labeling, Matching, Ontology Construction and Label Assignment. The Data wrapping method uses the CTVS module such as Tag Tree Construction, Data Region Identification, Record Segmentation, Query Result Section Identification, and Record Alignment.

The test sample is selected from TEL-8 dataset that UIUC University provides to perform the experiment. From the data set we selected three domains such as Automobile, and Books; in each domain query interfaces, the query condition has been typed. The query result pages are obtained manually. Suppose, if multiple pages are returned for a single query condition then the first page will be selected for our experiments. The collected web sites are manually classified in accordance with three categories listed as below:

(i) A Website has Multiple Record Pages (MRP)
(ii) A Website has Single Record Pages (SRP)
(iii) A Website has Empty Record Pages (ERP)

A. Data Set

There were totally 40 web sites collected from two domains such as Automobiles and Books, the statistics of data records distribution in these web sites were shown as table I. Since the typed query conditions were comparatively appropriate, there were less SRP and ERP in the collection. Moreover, there were more data records returned from Book domain, while less from Automobile domain; this phenomenon was actually determined by features of domain.

For each round, 10 out of the 20 Web sites for each domain are used as training Web sites to obtain the domain ontology and the remaining 10Web sites are test web sites. Since Training Web Sites does not required for CTVS and DeLa methods.

TABLE I. Statistical Data Set

Domain	No. Of Web Sites	No. Of MRP	No. Of SRP	No. Of ERP
Automobile	20	120	37	43
Book	20	155	18	27
Total	40	282	48	70

In this table 1, each web site contains number of query result record pages which are classified as follows: MRP, SRP, and ERP. Where, MRP represents multiple record pages which are relevant to the query condition. The number of SRP represents the Single Record page which has single relevant record to the query condition and other information such as advertisements, other information's etc., ERP represents the empty result page which is irrelevant to the query condition.

B. Performance measures

In order to evaluate our effectiveness of our data extraction method with attribute labelling we adopt precision and recall and F-Measure.

1) Precision

Precision is the ability to retrieve only relevant and discard the irrelevant records. Precision is the percentage of correctly annotated data values (Ca) over all annotated data values (Oa).

$$P = \frac{Ca}{Oa}$$

2) Recall

The recall value on the other hand expresses the number of retrieved relevant records out of all existing records. The recall is the ratio of correctly annotated data values (Ca) by over all data values (Od).

$$R = \frac{Ca}{Od}$$

3) F-Measure

F-Measure is the comprehensive evaluation of overall performance of precision and recall, and the larger value of f-measure is the better performance of annotation method. (i.e.) is used to measure the expected results based on the precision and recall.

$$Fm = \frac{2 * P * R}{P + R}$$

TABLE II. Performance Results for Data Annotation

Domain	Precision	Recall	F-Measure
Automobile	81.5%	78.5%	79.97%
Book	89.50%	86.50%	87.9%
Average	85.5%	82.50%	83.94%

The above table shows the values of precision , recall and the F-measure.

6. CONCLUSION

This paper has summarized in detail about the various existing data extraction and annotation methods. The web data extraction and annotation method and their drawbacks also have been discussed in the related work section. Among all other existing data extraction methods, Combining Tag and Value Similarity (CTVS) overcomes many of the drawbacks and gives high accuracy for the data extraction. The CTVS method is used to automatically extract and align the QRRs from a query result page. The data extraction using CTVS method yields more accurate precision and recall. But it still suffers from identifying an attribute in the query result table. To the enhancement of the existing technique, the label assignment (annotation) for the extracted query result record through the ontology domain is integrated to achieve the high performance. The overall performance is achieved by F-Measure metric using Precision and Recall.

REFERENCES

[1] Weifeng Su, Jiying Wang, and Frederick H. Lochovsky, "Combining Tag and Value Similarity for Data Extraction and Alignment", IEEE Transactions on knowledge and Data Engineering, Vol. 24, no.7, pp. 1186--1200, July 2012.

[2] Crescenzi V, Mecca G, and Merialdo P, "Roadrunner: Towards Automatic Data Extraction from Large Websites", In Proceedings of the 26th International Conference on Very Large Databases, pp. 109 –118, March 2001.

[3] D. Buttler, L. Liu, and C. Pu, "Omini-A Fully Automated Object Extraction System for the World Wide Web", In Proceedings of the 21st International Conference on Distributed Computing Systems, pp. 361-370, April 2001.

[4] A. Arasu and H. Garcia-Molina, "ExAlg: Extracting Structured Data from Web Pages", In Proceedings of the ACMSIGMOD International Conference Management of Data, pp. 337 - 348, July 2003.

[5] C.H. Chang and S.C. Lui, "IePAD: Information Extraction Based on Pattern Discovery", In Proceedings of the 10th World Wide Web Conference, pp. 681- 688, March 2001.

[6] J. Wang and F.H. Lochovsky, "DeLa: Data Extraction and Label Assignment for Web Databases", In Proceedings of the 12th World Wide Web Conference, pp. 187—196, July, 2003.

[7] J. Wang, and F. Lochovsky, "Data-Rich Section Extraction from Html Pages", In Proceedings of 3rd International Conference Web Information System Engineering", pp. 99-- 110, June 2002.

[8] D.W. Embley, and C.Tao, "TISP: Automatic Hidden-Web Table Interpretation by Sibling Page Comparison", In Proceedings of the 26th International Conference Conceptual Modeling, pp. 566--581, May 2007.

[9] Y. Zhai, and B. Liu, "Net – A System for Extracting Web Data from Flat and Nested Data", In Proceedings of 6th International Conference Web Information Systems Engineering, pp. 487-- 495, April 2005.

[10] Y. Zhai and B. Liu, "DePTA: Structured Data Extraction from the Web Based on Partial Tree Alignment", IEEE Transactions on Knowledge and Data Engineering, vol.18, no.12. pp. 1614—1628, December 2006.

[11] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, "ViNTs - Fully Automatic Wrapper Generation for Search Engines", In Proceedings of the 14th World Wide Web Conference, pp. 66--75, May 2005.

[12] K. Simon and G. Lausen, "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions", In Proceedings of the 14th ACM International Conference Information and Knowledge Management, pp. 381—388, June 2005.

[13] C Tao, and D.W.Embley, "TISP++: Automatic hidden-Web table interpretation, Conceptualization, and semantic annotation", IEEE Transactions on knowledge and Data Engineering, vol. 68, no.7, pp. 683--704, July 2009.

[14] Lu, Y., Hai He., Zhao. H, Meng. W, and Yu.C, "Annotating Structured Data of the Deep Web", In Proceedings of the 23rd IEEE International Conference on Data Engineering, pp. 376–385, April 2007.

[15] A. Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages", In Proceedings of the ACM SIGMOID International Conference Management of Data, pp. 337--348, March 2003.

[16] B. Liu, R. Grossman, and Y. Zhai, "Mining Data Records In Web Pages", In Proceedings of the 9th ACM SIGKDD International Conference Knowledge Discovery and Data Mining", pp. 601--606, December 2003.

[17] H. Snoussi, L. Magnin, and J.-Y. Nie, "Heterogeneous Web Data Extraction Using Ontologies", In Proceedings of 15th International Conference Agent Oriented Information Systems, pp. 99-- 110, April 2001.

[18] X. J. Cui, Z. Y. Peng and H. Wang, "Multi-Source Automatic Annotation for Deep Web", In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, pp. 659 -- 662, May 2008.