International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.2, Issue No.6, pp : 573-578 1 June 2013

Web Data Extraction and Alignment Tools: A Survey

Shridevi A. Swami, Pujashree Vidap

Department of Computer Engineering, Pune Institute of Computer Engineering Pune, India.

shrideviswami@gmail.com

Abstract**—*****Search engine generates the dynamic result page when user submits a query. Result page consists of query relevant data along with some auxiliary information such as advertisement, navigation panels. Decision making regarding which part of this web page has main content is easy for human but tough for computer programs. So in order to utilize this data, it is necessary to remove irrelevant data and automatically extract data from those result pages. Further extracted data can be aligned in structured format like table for comparison.***

This paper deals with the study of various automatic web data extraction and data alignment techniques. Web data extraction techniques are mainly classified as Wrapper programming languages, Wrapper induction and Automatic extraction. For data alignment some techniques rely only on structure of html tags or on both tag and data values.

Keywords—**Data extraction, Wrapper induction, DOM tree, Web crawler, Data alignment**

I. INTRODUCTION

World Wide Web is a powerful source of information. Search engines are very important tools for people to get the desired information on the web. Not only web users but many web applications also need to interact with search engines.

For decision making many business applications have to depend on web in order to aggregate information from different web sites. By analyzing and summarizing web data we can find latest market trends, price details, product specification etc. Manual data extraction is time consuming and error prone. 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.

Automatic data extraction plays an important role in processing results provided by search engines after submitting the query by user. Wrapper is an automated tool which extracts Query Result Records (QRRs) from HTML pages returned by search engines. Automated extraction is easier with the sites having web service interfaces like Google and Amazon. But it’s difficult for those that support B2C i.e. business to customer applications which does not have web service interfaces. Normally Search engine result consists of query independent contents (static contents), query dependent contents (dynamic contents), while some contents are affected by many queries but independent of content of specific query (semi-dynamic). As the web evolved web page creation process changed from manual to a more dynamic procedure using complex templates. Many web pages are not created in advance, but are generated dynamically by querying a database server and sending the results to a predefined page structure. Automatic data extraction is very important for many applications, such as meta-querying, data integration and comparison shopping, that need to co-operate with multiple web databases to collect data from multiple sites and provide services.

This paper gives the overview of various information extraction techniques like DeLa[1], DEPTA[2], ViPER[4], ViNTs[9]. Section II discusses each technique in detail, section III provides a comparison and section IV concludes the paper.

II. WEB DATA EXTRACTION AND ALIGNMENT TOOLS

Mainly web data extraction can be classified into three categories: 1) Wrapper programming languages, 2) Wrapper induction, and 3) Automatic extraction.

1) Wrapper programming languages

This approach uses the special pattern specification languages which help the user to develop extraction programs. Visual platforms are also provided to hide their complexities under simple graphical wizards and interactive processes.

Examples:-Systems that use this approach include WICCAP, Wargo, Lixto, etc.

2) Wrapper induction method

This method is useful in systems where the resource information is formatted for use by people and so it is difficult to extract their content mechanically. So a technique for constructing wrappers automatically from labelled examples of a resource’s content is introduced called wrapper induction. This approach uses the extraction rules which are derived by using inductive learning. In these methods there is requirement of human assistance for building a wrapper. The user labels the items present in a set of training pages or in a list of data records on page, which are to be extracted as target items. Then the system learns the wrapper rules from the labelled or marked data and further uses them to extract records from new pages. The wrapper rule is made up of two patterns. One is prefix pattern which denotes the beginning of the target item and other is suffix pattern which denotes the end of the target item.

Examples:-Some existing systems that uses wrapper induction include WIEN, Soft Mealy, S talker, XWRAP, WL and Lixto.

Advantage:- As user itself labels the items of interest, no extra data are extracted.

Disadvantages:-

- Manual labelling of data is time-consuming.
- It is not scalable to a large number of web databases.
- Existing wrapper gives poor performance when the format of a query result page changes, which happens more frequently on the web.
- Continuous monitoring is needed to keep track on changes in format of pages and maintaining a wrapper when a page's format changes.

3) Automatic extraction methods

To overcome the problems of wrapper induction, some unsupervised learning methods, have been proposed to automatically extract the data from the query result pages. These methods rely entirely on the tag structure in the query result pages.

A. *DeLa (Data Extraction and Label Assignment for Web Databases)*

DeLa system automatically extracts the data from web site and associates meaningful labels to data. Complex search forms are used by this method rather than using keywords to keep track on pages that querying back end database. Fig. 1 shows the architecture of DeLa.

DeLa is a System that automatically extracts text from a web-page into a table and assigns labels in a table with the help of four components i.e. a form crawler, wrapper generator, data aligner, label assigner.

Form crawler: It collect labels of the website form elements. Hidden web crawler HiWe [5] is used for this purpose in DeLa. Data present in pages is used by wrapper generator for automatic generation of regular expression wrappers.

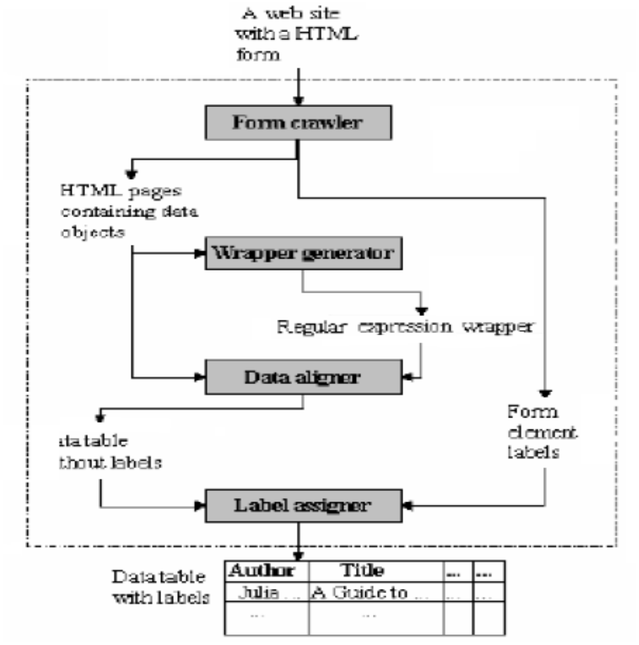

Fig. 1 DeLa Architecture

The characteristics and semantics of the element are understood with the help of text contained in form elements. So descriptive text is used for labelling form elements, which are useful to do further comparison with attributes of data extracted from query-result page.

Wrapper Generation: The input to the wrapper generator is pages collected by the form crawler. Wrapper generator produces regular expression wrapper based on HTML tag structures of the page. If a page contains more than one instance of data objects then tags enclosing data objects may appear repeatedly. Wrapper generator considers each page as a sequence of tokens composed of HTML tags. Special token "text" is used to represent text string enclosed within HTML tag pairs. Wrapper generator then extracts repeated HTML tag substring and introduces a regular expression wrapper according to some hierarchical relationship between them.

Techniques used in wrapper generator are

i) Data-rich section extraction

ii) C-repeated pattern

iii) Optional attributes and disjunction

Data Alignment : Data aligner works in two steps i.e. data extraction and attribute separation.

i) Data exaction

This step does data extraction from web pages by using the wrapper produced by wrapper generator. Then the extracted data will be loaded into a table. In data extraction step regular expression pattern and token sequence is used for representing web page. A nondeterministic finite automation is constructed to match the occurrence of token sequences representing web pages. A data-tree will be constructed for each regular expression.

ii) Attribute separation

Prerequisite for attribute separation is the removal of all HTML tags. If several attributes are appearing in to one text string then they should be separated by special symbol(s) as separator. Some examples of invalid separators are "@", "$", ".". If multiple separators are found to be valid for one column, then attribute strings of this column are separated from beginning to end in the order of occurrence portion of each separator.

Label Assignment: To assign labels to the columns of the table which contain extracted data following four heuristics are used[1]:

Heuristic 1: Match from element labels to data attributes.

Heuristic 2: Search for voluntary labels in table header

Heuristic 3: Search for voluntary labels encoded together with data attributes.

Heuristic 4: Label data attributes in conventional formats.

Drawbacks of DeLa:-

1. DeLa often produces multiple patterns (rules) and it is hard to decide which is correct.

B. *DEPTA (Data Extraction based on Partial Tree Alignment)*

DEPTA is a system which performs automatic data extraction given a single page with lists of data records. The general architecture of DEPTA is shown in fig 2.

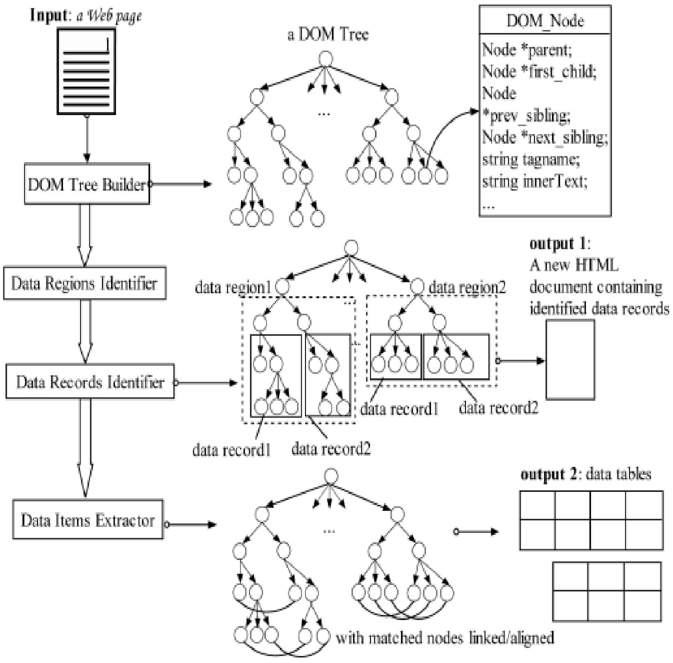

Fig. 2 DEPTA Architecture

Web page containing lists of data records is given as input to the DEPTA system. The system is composed of the following main components.

a. Building HTML tag tree (DOM Tree)

Tag tree is constructed as follows.

- Find four boundaries of rectangle of each HTML tag by calling embedded parsing and rendering engine of browser [3].
- Detect containment relationship between rectangles. Then construct tag tree based on containment relationship.

b. Mining data region

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 labeled 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 gap between the data records in a data region should be no smaller than any gap with in a data record [2].

c. Identifying data records

Data records are identified from generalized nodes. There are two cases in which data records are not in contiguous segment:

Case1 : Data region contains two generalized nodes each of which contains two tag nodes which indicate that they are no similar to each other. Each node has same number of children which are similar to each other.

Case 2: Two or more regions form multiple data records .

d. Data item extractor

It is performed based on partial tree alignment technique to match corresponding data item or fields from all data records. Two sub-steps are

- Production of one rooted tag tree for each data record. Subtrees of all data record 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 are involved in matching process. Only tag nodes are used for matching. Tree edit distance between two trees is cost associated with minimum set of operations needed to transform A in to B. Restricted matching algorithm called simple tree matching is used which will produce maximum matching between two trees.

Multiple tag trees of multiple data records are needed to align in order to produce a data base table. In this data table each row represents a data record and column represents data field. This can be performed using multiple alignment method. Partial tree alignment is used in DEPTA. This approach aligns multiple tag trees by progressively growing a seed tag tree. Seed tree is the tree with minimum number of data fields that is picked initially. The selection of seed tree should be in such a way that it should have a good alignment with data fields in other data records. For each tree $T_i[i \neq s]$ the algorithm tries to find a matching node in T_s. When a match is found for node n_i, a link is created from n_i to n_s to indicate its match in the seed tree. If no match found then algorithm attempts to expand the seed tree by inserting n_i in to T_s. The expanded seed tree is used for subsequent match.

C. ViPER (Visual perception based Extraction of Records)

Deriving accurate wrappers based solely on HTML tags is very difficult for the following reasons [7].

1) One cannot fully depend on “proper” HTML tag usage since HTML tags are often used in unexpected and unconventional ways.

2) The main purpose of HTML tags is to facilitate the rendering of data so they convey little semantic information.

3) The data containing embedded tags may confuse the wrapper generators in turn making them less reliable.

To overcome these problems, methods such as ViPER and ViNTs make use of additional information in the query result pages.

ViPER 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 visible similarity. ViPER extracts relevant data with respect to user’s visual perception of the web page. Then Multiple Sequence Alignment (MSA) method is used to align these relevant data regions.

ViPER uses both visual data value similarity features and the HTML tag structure to first identify and rank potential repetitive patterns. Then, matching subsequences are aligned with global matching information. But ViPER suffers from poor results for nested structured data.

ViPER is a two step process i.e. Data Extraction and Data Alignment.

i) Data Extraction

HTML document can be viewed as labelled unordered trees. A labelled unordered tree is a directed acyclic graph $T = (V, E, r, \eta)$ where V is set of vertices, E is set of edges, R is root and η is the label function η: V X L where L is a string.

Data Extraction consists of following substeps:

Preprocessing: It is used to improve pattern extraction accuracy. Preprocessing provides the ability to access parsed document tree T* with additional rendering information. Every tag element is augmented with bounding box information by the upper left corner’s (x,y) pixel coordinates along with width and height. For analysis abstract representation of T* is created in which each HTML tag is restricted to tag name ignoring attributes. Text between two tags represented by a new element denoted as <TEXT> element tag. Preprocessed document is called restricted tag tree T and plain tag sequence structure S of T where each element in the tree has a link to the corresponding element in the sequence representation and vice versa [4].

Pattern search : Similarity between two plain sequences S_i, S_j with length m, n respectively is measured using technique edit distance. One disadvantage with edit distance is that repetitive and optional subparts inside the sequence S_i, S_j should increase edit cost, so possible matches may be discarded. These optional subparts are handled by similarity threshold value θ. Two sequences will be similar if their accumulated edit distance is less or equal to threshold value.

Primitive tandem repeats : Tandem repeat contained in a sequence is a subpart of S. Tandem repeat construct an array of consecutive repeats. A repeat is primitive if it does not contain shorter repeats. Each extra repetitive instance will be marked with different marker elements. According to these marked tag elements the recursive computation of a single matrix entry of D is adapted[8].

Identifying data regions and record: Single data records with in a data region may consist of variable number of subtrees. When computing pair wise similarity between all subtree sequences produce an upper triangular subtree similarity matrix M_u for each inner node u. To simplify pattern discovery edit-distance values are not stored inside the matrix. Cell entry $M_u(i, j)$ becomes 1 if the edit distance between two sequences S_{vi}, S_{vj} satisfies specific conditions. Next to identify sets of adjacent sibling nodes having highest matching frequency.

Visual Data Segmentation: After identifying data region R_k, the corresponding image representation defined by R_k is analyzed. Bounding boxes of <TEXT> elements contained in R_k are used to compute vertical and horizontal projection profiles. This is realized by summing the width and respective height of all boxes with respect to their x/y coordinates[4].

In x-y profile information valleys between peaks corresponds to blank between text lines and distance between two significant valleys corresponds to potential separation of data region into smaller data records. Relationship between valleys and tag elements are established by containment check.

Visual data region weighting: After pattern extraction, the next step is measure the relevance of each pattern. Several heuristics are used to measure individual weight of a region. One heuristics is to compute the textual coverage of data region using pattern size. But this fails if pattern contain images, links or the region that has highest textual coverage. Vision-based page segmentation can be performed by ranking technique, where features of a pattern are determined by visual location on the page. Web pages are divided into different information regions called slots. Slot filled with target information has often a centre location and covers large part of the page.

ii). Data Alignment

Global sequence alignment uses general suffix tree. Global data record alignment algorithm try to find maximal matches (MUMs) contained in all records. Maximal means for every sequence we cannot extend a match to left or right and unique means match occurs only once in each n sequences. MUM sequences of non overlapping MUMs with maximal weight are selected. Weight of MUM sequence is the sum of weights of its constituting elements. If we are trying to align data records using MUM sequence of length l, the problem decomposes into l+1 smaller unaligned sub regions. Each sub region is iteratively aligned separately using global matching.

In case of text alignment, it contains building general suffix tree and checking regularities contained in all sequences. Two abstract <TEXT> elements A and B are content similar if they are similar w.r.t their original trimmed text content.

Drawbacks of ViPER:-

1. ViPER suffers from poor results for nested structured data.

*D. ViNTs (**V**isual **i**nformation a**N**d **T**ag **s**tructure based wrapper generator)*

It is a tool for automatically producing the wrappers which extract the Search Result Records (SRRs) from dynamically generated HTML result pages returned by any search engines.

ViNTs [9] 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. The final resulting wrapper is represented by a regular expression of alternative horizontal separator tags (i.e., <HR> or <BR> <BR>), which segment descendants into QRRs.

The architecture of ViNTs system is as shown in fig 3. 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.

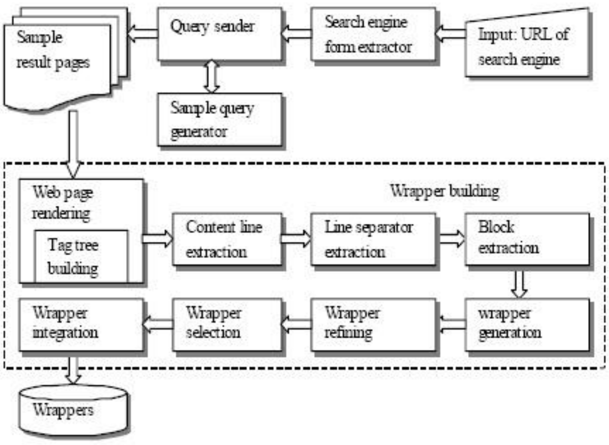

Fig.3. ViNT Architecture

Drawbacks of ViNTs:-

1. Several result pages, each of which must contain at least four QRRs, and one no-result page are required to build a wrapper.

1. If the data records are distributed over multiple data regions only the major data region is reported.

2. 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.

3. The prelearned wrapper usually fails when the format of the query result page changes.

Hence, it is necessary for ViNTs to monitor format changes to the query result pages, which is a difficult problem.

E. CTVS (Data Extraction and Alignment using Combining Tag and Data Value Similarity)

CTVS (Combining Tag and Value Similarity) [8] a novel approach that uses both 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 same column.

The novel aspect of this method is that mainly it handles two cases, first when QRRs are not contiguous, as 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.

CT VS (CTVS) consists of following two-steps, to extract the QRRs from a query result page P.

1. Record extraction

It identifies the QRRs (Query Result Records) in P and involves following steps:

a) Tag tree construction
b) Data region identification
c) Record segmentation
d) Data region merge
e) Query result section identification

2. 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.

IV. Conclusions

This paper discussed various approaches to extract structured data from web pages. We can summarize these web data extraction methods as follows:

Among the above discussed web data extraction methods, some techniques reveals flat records and some other techniques are trying to extracts nested records also. DEPTA and DeLa will find out nested records in addition to flat records. ViPER is not able to handle nested structured data.

DeLa, ViPER and ViNTs are not able to handle non contiguous data regions, while CTVS can handle both non contiguous and nested structure data.

DeLa, extracts records using wrapper induction method, others are based on operations on tree structure of the page such as tree alignment, tree merging and tree matching. In DEPTA extraction is performed mainly by partial tree alignment. ViPER uses the extraction method which is based on visual perception.

DeLa, ViPER and CTVS consider single result web page while ViNTs considers multiple pages of web site.

References

1. J. Wang and F.H. Lochovsky, “Data Extraction and Label Assignment for Web Databases,” Proc. international conference on World Wide Web (WWW-12), pp. 187-196, 2003.
2. Y. Zhai and B. Liu, “Web Data Extraction Based on Partial Tree Alignment,” Proc. international conference on World Wide Web (WWW-14), pp. 76-85, 2005.
3. A. Arasu and H. Garcia-Molina, “Extracting Structured Data from Web Pages,” Proc. ACM SIGMOD, pp. 337-348, 2003.
4. K. Simon and G. Lausen, “ViPER: Augmenting Automatic Information Extraction with Visual Perceptions,” Proc. International Conference on Information and Knowledge Management (CIKM), 2005.
5. S. Raghavan and H. Garcia-Molina. "Crawling the hidden web," Proc. 27th VLDB Conf., 2001, 129-138.
6. J. Wang and F. Lochovsky. "Data-rich section extraction from HTML pages," Proc. 3rd Conf. on Web Information Systems Engineering, 2002, 313-322.
7. Y. Zhai and B. Liu, “Structured Data Extraction from the Web Based on Partial Tree Alignment,” IEEE Trans. Knowledge and Data Eng., vol. 18, no. 12, pp. 1614-1628, Dec. 2006.
8. W. Su, J. Wang, F. H. Lochovsky, and Yi Liu, " Combining Tag and Value Similarity for Data Extraction and Alignment ", IEEE Trans. Knowledge and Data Eng., vol. 24, no. 7, pp. 1186-1200, July. 2012.
9. H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, “Fully Automatic Wrapper Generation for Search Engines,” Proc. 14th World Wide Web Conf., pp. 66-75, 2005.