ISSN 2277-3061

Towards Automatic Web Data Scraper and Aligner (WDSA)

Shridevi A. Swami, Pujashree S. Vidap

Pune Institute of Computer Technology, Pune University, Maharashtra, India
shrideviswami@gmail.com
Pune Institute of Computer Technology, Pune University, Maharashtra, India
psvidap@pict.edu

ABSTRACT

Web is very immense and fast emerging source of information. Web browsers along with search engines have come forward as famous tools for retrieving and accessing the information present on web. Enormous growth of web made the data extraction from web harder than ever. This paper presents the Automatic Web Data Scraper and Aligner (WDSA). Automatic WDSA extracts the interested web data present in dynamically generated web page received from search engine when user gives a query. Automatic web data scraping is necessary because human being can identify the interested query relevant contents from query result web page, however it is tricky for computer applications. Extracted web data can be further transferred into a format suitable for use in applications like comparison shopping, data integrations, value added services etc. WDSA does this by aligning the extracted web data pairwise as well as holistically in table. The novel thing about Automatic WDSA is that Data Scraper and Aligner uses new approach which combines similarity of both tag and value, for extraction and alignment process. Also Data Scraper handles the data which is present in non contiguous fashion due to presence of auxiliary information like advertisement banners, navigational links, pop ups etc. Experimental results show that Automatic WDSA achieves high precision and recall. Further Automatic WDSA is compared with existing most widely used famous tools like Helium scraper, Outwit Hub, Screen Scraper etc. During comparison we observed that Manual labeling or extraction patterns of desired data is to be specified for working of existing tools while Automatic WDSA does not require any user involvement which made it fully automatic.

Indexing terms/Keywords

Data extraction, Wrapper, Data scraping, Data values alignment, Data integration.

Academic Discipline And Sub-Disciplines

Knowledge and data engineering

SUBJECT CLASSIFICATION

Computer Science

TYPE (METHOD/APPROACH)

Web scraping ; Web data alinment

INTRODUCTION

Currently the quantity of information existing on the net in HTML format grows at a very fast rate. This makes the web available to the public as largest “knowledge base” ever developed. However HTML sites can be referred as modern legacy systems, because such a huge data cannot be accessed and manipulated easily. The reason for the same is that web data sources are to be browsed by human beings, and not computed over by applications. This leads to the consequence such as web pages data extracting process and making its availability to computer applications becomes a relevant and complex job.

Software units called wrappers are mainly used for data extraction from HTML pages. The foundation of early approaches towards the wrapping of web sites was on manual techniques [12], [13], [16] and [17]. A key dilemma with manually implied wrappers is that coding them is usually a complicated and manual effort demanding task, and further wrappers tend to be fragile and difficult to maintain by their nature. To overcome with these problems some unsupervised learning methods [3], [5], [7], [10] and [14] have been proposed to automatically extract the data from web pages. Such methods are fully using the tag structure of HTML pages which may lead to inaccurate extraction.

This paper provides spotlight on the problem of automatically extracting user query relevant data records that are encoded in the query result pages generated dynamically by web databases.

Generally, in response to a user query when submitted through the query interface of a web database, deep web generates dynamic web pages unlike in surface web where unique URL is used to access the web pages. On receipt of a user’s query, a web database gives the relevant data, encoded in HTML web pages in either structured or semi structured format. Many web applications, like comparison shopping, meta querying and data integration or aggregation require the data to be supplied from various web databases. These applications need to exploit the data which is embedded in HTML pages, which further leads to conclusion that automatic data extraction is crucial. Further everyone knows the very well known fact that when the data are extracted and structured in a well organized fashion, like tables, only then they can be aggregated and compared. Hence, accurate data extraction or scraping is the main central point of attention for these applications to achieve high accuracy by performing correctly.

This paper contributes to the development of a new approach to the web data extraction problem by fully automating the wrapper generation process, by making it independent of any prior knowledge regarding the target pages and their contents.

In universal, a query result page contains not only the actual data, but also other decorative information, such as advertisements, navigational panels, comments, information related to sites used for hosting, and so on. The aim of web data extraction is to remove any irrelevant information from the query result page, extract the Query Relevant Records i.e. QRRs the web page, and further align the extracted QRRs into a table such that the data values belonging to the same attribute are placed into the same table column.

Automatic WDSA uses two novel steps, QRR scraping or extraction and QRR alignment to extract the records from query result page and then align the data values of records into table.

1. QRR scraping or extraction: - This step identifies the QRRs in query result page by identifying the data regions and further segmenting it into records.
2. QRR alignment: - This step aligns the data values of QRRs in table, pairwise and holistically such that the data values for same attribute will share the same column of table.

The rest of the paper is organized as follows: Section 2 reviews recent work on data extraction. Section 3 describes the system architecture for Automatic WDSA along with main steps of our method: Data Scraping and Data alignment. Section 4 shows result screen shots of implementation. Section 5 describes performance evaluation our method. Finally section 6 concludes the paper.

RELATED WORK

An interesting active research field for many years is improving the wrapper generation techniques used for extraction of web data. Till now many approaches have been proposed and [14] provides comparison of surveyed techniques.

All the wrapper generation methods require some kind of human involvement to build and constitute the wrapper. However in applications where the sources are unknown in advance, this approach will not be feasible. So, automatic wrapper generation techniques are introduced. Several works have addressed the problem of performing web data extraction tasks without requiring human input.

IEPAD [3] uses the techniques such as Patricia tree and string alignment to search the HTML tag string of a page to find repetitive patterns. However there is high probability that the method used by IEPAD generates incorrect patterns along with the correct ones, so human involvement is required for post-processing of the output.

RoadRunner [15] gets multiple pages conforming to the same template as input and union-free regular expression (UFRE) is induced from them which can be further used to extract the data from the pages conforming to the template. The basic idea behind the RoadRunner is performing an iterative process where the system takes the first page as initial UFRE and then, for each subsequent page, tests if it can be generated using the current template. If not, the template is modified to represent the new page. The drawback of the proposed method is that it requires multiple pages conforming to the same template as input and further it is unable to deal with disjunctions in the input schema.

DEPTA [2] uses the details about visual layout of information present in the page and tree edit-distance techniques to identify lists of records in the page and further extracts the structured data records that make the page. DEPTA requires can receive one single page containing a list of structured data records as input and uses the observation that, DOM tree of a page consists of a set of consecutive sibling subtrees which generates each record in a list.

On the other hand, following additional assumptions are made:

1) All records must be formed by exactly the same number of sub-trees, and

2) The visual gap between two data records in a list is bigger than the gap between any two data values from the same record. However, in all web sources these assumptions do not hold.

DeLa [7] models the structured data present in template-generated web pages as string instances which are encoded in HTML tags, of the implied nested type of their web database. A regular expression is used to model the HTML-encoded version of the nested type. If the page contains more than one instance of the data then the HTML tag-structure enclosing the data appears repeatedly. So in this case the page is first changed into a token sequence collected of HTML tags and a special token “text” used for representing text string which is enclosed by pairs of HTML tags. Then, continuous repeated substrings are extracted from the token sequence and a regular expression wrapper is induced from the repeated substrings according to some hierarchical relationships present among them. The main problem with this method is that it often produces multiple patterns (rules) and it is hard to decide which one is correct.

Since deriving the accurate wrappers based exclusively on HTML tags is very difficult [2] some more techniques are introduced which use the additional information from query result page.

ViPER [10] 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.

ViNTs [5] learns a wrapper from a set of training pages from a website by using both visual and tag features. 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. Several result pages, each of which must contain at least four QRRs, and one no-result page are required to build a wrapper.

Among the above discussed web data extraction methods, some techniques reveals flat records while some are not able to handle non contiguous data regions. Also some techniques require training pages along with a prelearned wrapper for a website. 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. Most of the wrappers are based exclusively on HTML tags structure.

In contrast, Automatic WDSA uses similarity of both tag with its data value and also handles non contiguous data regions. Further it requires neither training pages nor a prelearned wrapper for a website.

SYSTEM ARCHITECTURE

Automatic WDSA is designed with the objective as to automatically extract the Query Relevant Records (QRRs) in a page, and align the data values of the QRRs into a table. Automatic WDSA architecture is shown in Figure 1.

The system takes the input query result page containing atleast two QRRs decorated with auxiliary information. The query result page passes through two phases i.e. Data scraping and Data alignment to give output as data records referred as QRRs and aligned data values respectively. Data scraping phase consists of four steps, DOM tree builder, Data region identification, Record segmentation and Query result section identification. Data alignment consists of two steps, Pairwise alignment and Holistic alignment.

When a query result page is given as input, the tag tree for the page rooted in the <HTML> tag is constructed by DOM Tree Builder module. Next, in the Data Region Identification module, identification of all possible data regions is done, which usually contain dynamically generated data, top down starting from the root node. When system enters into the Record Segmentation module it segments the identified data regions into data records according to the tag patterns in the data regions. Finally, when the segmented data records are given, the Query Result Section Identification module is responsible for selecting one of the data regions as the one that contains the QRRs. Further when QRRs are supplied to Pairwise Alignment module, aligns the data values of QRRs which further holistically aligned by Holistic Alignment module.

ISSN 2277-3061

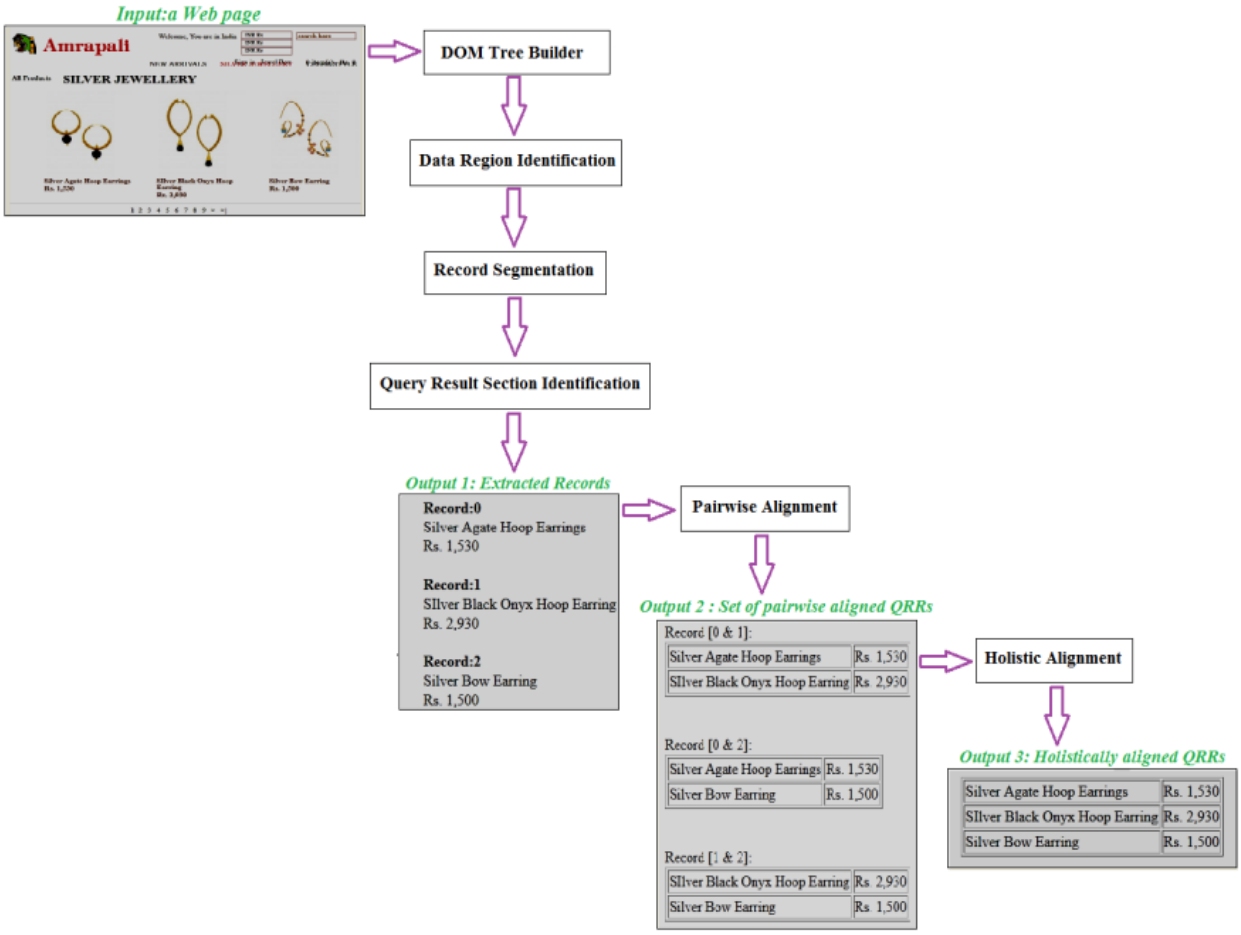


Fig. 1 System Architecture

DATA SCRAPING OR EXTRACTION

DOM Tree Builder (Tag Tree Construction):-

We are concerned in detecting and extracting the Query Relevant Records (QRRs) embedded in HTML pages. HTML pages can be represented as DOM or Tag tree. The representation of pages as DOM trees exhibit the following properties:

1) Property 1:- Each QRR in DOM tree is presented in a set of consecutive sibling subtrees.

2) Property 2:- The occurrences of each attribute in several QRRs share the same path from the root in the DOM tree.

For given input Query result HTML page Tag Tree Construction module builds the Tag or DOM tree rooted in <html> tag. Every node represents a tag in html page and its children are tags enclosed inside it. Each internal node 'n' of the tag tree has a tag string 'ts_n', which includes the tags of 'n' and all tags of n's descendants.

Data Region Identification:-

According to property 1, each QRR is composed of one or more consecutive sibling subtrees sharing same parent node which are direct descendants of the root node of the data region. Here we propose new data region identification algorithm for handling QRRs that can be present in non contiguous region. Given query result page containing minimum two QRRs, data region identification algorithm finds data regions in a top down fashion.

Algorithm:

/* DataRegionIdentification procedure finds data regions by applying following steps recursively to children of every node in T only if it does not have similar siblings.*/Proc DataRegionIdentification (Tag Tree T)

1. Calculate similarity sim_{ij} of each pair of nodes n_i and n_j, i,j=1…m and i≠j, using similarity computation method.

2. Group the nodes according to their similarity and assign them respective sibling identifier, sib_id. Finally before iteration completes the grouped nodes sharing the same parent are clustered under same data region identifier. As shown in Figure 2 first and third TR nodes get the sib_id as 1, second and forth TR nodes get the sib_id as 2. Further these nodes are clustered under Region 1. Similarly Region 2 consists of two TD nodes with sib_id 3.

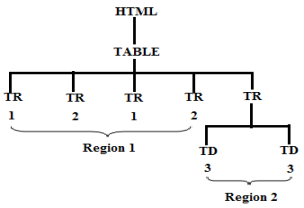


Fig. 2 Artificial Tag Tree

Similarity computation method [6]:- Two nodes are similar if their similarity is larger than or equal to threshold, which is set to 0.6. Tag strings of nodes are considered to calculate the node similarity. Use following dynamic programming approach for calculating edit distance similarity between nodes n_i and n_j with tag string ts_i and ts_j which computes the desired final solution $ed(n_i,n_j)$ by filling incrementally an (m + 1)X(n + 1) dynamic programming table D, in which each cell D[i,j] will eventually hold the value $ed(ts_{1..i}; ts_{1..j})$ for i = 0..M and j = 0..N.

D[m,0]=m ; D[0,n]=n

If $ts_m=ts_n$, D[m,n]= D[m-1,n-1]

Otherwise, D[m,n]=1+min(D[m-1,n-1],D[m-1,n], [m,n-1]),

Where m=0…M,n=0…N, M and N are length of ts_i and ts_j respectively.

For getting more accuracy normalized edit distance is considered which is calculated as,

$$Ned(n_i,n_j) = 1 - ed(n_i,n_j) / (length(ts_i) + length(ts_j))$$

Record Segmentation:-

This module divides the identified data region into set of possible records based on pattern governed by region. For segmenting the region into records, we build a sequence by listing in order the nodes in the data region, representing each node with the sib_id assigned to it. For example, referring the fig 1, algorithm generates the sequence for Region 1 as 1212 and for Region 2 as 33.

By property 1, we know each QRR or record is formed by a list of consecutive subtrees i.e. records are encoded consistently. Therefore the pattern will tend to be formed by tandem repeats i.e. repetitive sequence of sib_id, each sequence corresponding to a QRR. So, in Region 1, tandem repeats 12 correspond to record and in Region 2 tandem repeat 3 correspond to record.

Query Result Section Identification:-

Data region identification module identifies multiple data regions in the input query result web page. But we assume that only one data region consists of QRRs. So following three rules are used to find this final region called as Query Result Section.

1. Data region with the largest area in the query result page is Query Result Section.
2. Data region located at the center of the query result page is Query Result Section.
3. Data region with more data strings than others is Query Result Section.

Data region satisfying above three rules is selected as Query Result Section and assumed that it contains the required QRRs.

Data Alignment:-

QRRs are aligned with the help of two steps.

1. Pairwise QRR Alignment:- Here data values of pair of QRRs are aligned such that data values of same attribute are aligned in same column.
2. Holistic QRR Alignment:-Here all the QRRs are aligned globally.

Pairwise QRR Alignment:-

Pairwise QRR Alignment algorithm uses the observation that data values of same attribute are usually of same data type and may composed of similar strings, because QRRs are for the same query.

Algorithm:-

Proc PairwiseAlignment (set of QRRs)

Input: Pair of QRRs, $QRR_1 = \{d_{11},\ldots, d_{1m}\}$ where d_{1i} refers to i^{th} data value of QRR_1, $QRR_2 = \{ d_{21},\ldots, d_{2n} \}$ where d_{2i} refers to i^{th} data value of QRR_2

Output: Aligned pairs of QRRs presented as tables

Constraints: a) Each data value can be aligned to at most one data value from the other QRR.

b) Cross alignment cannot be performed.

Steps:

1. Find data value similarity 's' between d_{1i} and d_{2j} based on following rules which use the Data Type (DT) information of the data values.

   If (DT (d_{1i}, d_{2j}) = Interger) then s=1;

   If (DT (d_{1i}, d_{2j}) = Double) then s=1;

   If (DT (d_{1i}, d_{2j}) = Price) then s=1;

   If ((DT (d_{1i}) = Integer) AND (DT (d_{2j}) =Double) OR

   (DT (d_{1i}) =Double) AND (DT (d_{2j}) =Integer)) then s=0.5;

   If (DT (d_{1i}, d_{2j}) = String) then s=cosine similarity (d_{1i}, d_{2j})

   If (DT (d_{1i}) != DT(d_{2j})) then s=0;

2. Based on constraints two data values of two QRRs with largest similarity score will be aligned. Similarity score will be calculated by using following dynamic programming equation.

   L_{ij} = Max ($L_{(i-1)(j-1)}$+s_{ij}, $L_{(i-1)j}$, $L_{j(i-1)}$)

   Here s_{ij} is data value similarity calculated in step 1, i and j are number of data values in QRR_1 and QRR_2 respectively.

Holistic QRR Alignment:-

Pairwise data value alignments between every pair of QRRs is provided to this Holistic alignment step which performs the global alignment among all QRRs and presents it in table format where all data values of same attribute are placed in same table column.

Algorithm:

Proc HolisticQRRAlignment ()

Input: QRRs and Pairwise aligned table for each pair of QRRs

Output: Holistically aligned QRRs shown as global table

Steps:

1. Represent the pairwise alignment tables as graph by assuming each data value in the QRRs as vertex and each pairwise alignment between two data values as an edge.
2. Find the connected components present in graph by traversing the graph in depth first search manner.
3. Check the discovered connected components against following constraints.

a. No vertices from same connected component should belong to same QRR, since they are coming from two different attributes of the same QRR.

b. No intersection is allowed for connected components.

4. Each connected component of the graph represents a table column inside which the connected data values from different QRRs are aligned vertically.

IMPLEMENTATION AND RESULT

Automatic WDSA system was implemented in JAVA on Intel i3 2.20GHz CPU with 1 GB memory. This section discusses the results generated for input query result page taken from data set 2 shown in Figure 3.

ISSN 2277-3061

Fig. 3 Sample Input Web Page

The Figure 4 shows the entry page of Automatic WDSA system where user has to select the input query result page who's QRRs are to be extracted and aligned in table.

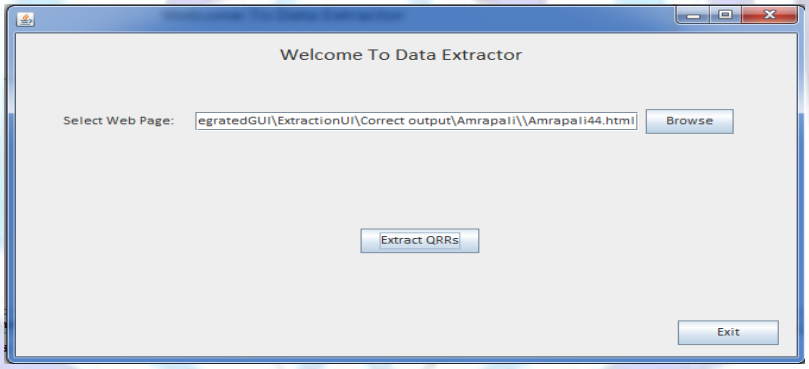


Figure. 4 Selection of query result page

The tag tree for selected query result page is shown in Figure. 5.

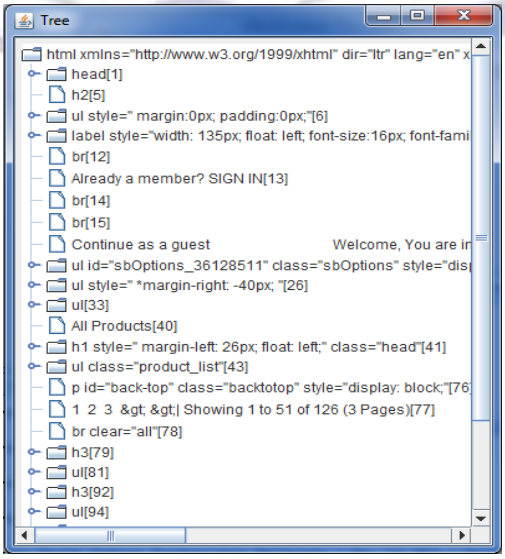


Fig. 5 Tag or DOM Tree

The Figure 6 shows result for data scraping steps i.e. for data region identification, record segmentation and query result section identification steps.

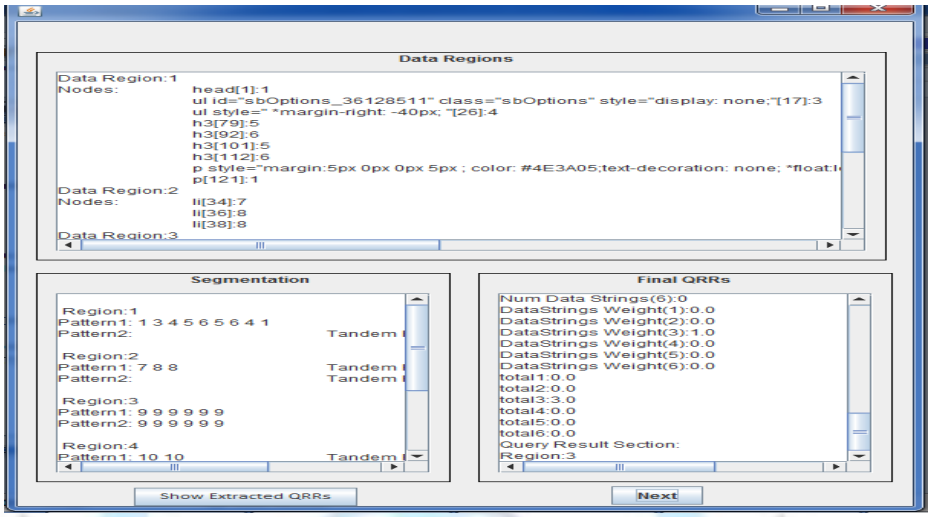
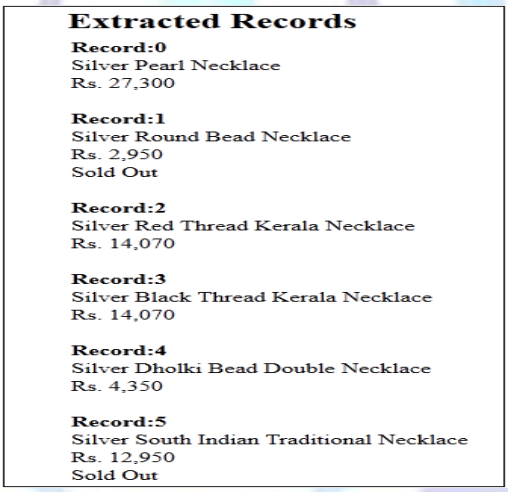


Fig. 6 Result for Data Scraping Steps

The Figure 7, 8 and 9 show the extracted QRRs, pairwise alignment of QRRs and holistic alignment of QRRs respectively.



Extracted Records

Record:0
Silver Pearl Necklace
Rs. 27,300

Record:1
Silver Round Bead Necklace
Rs. 2,950
Sold Out

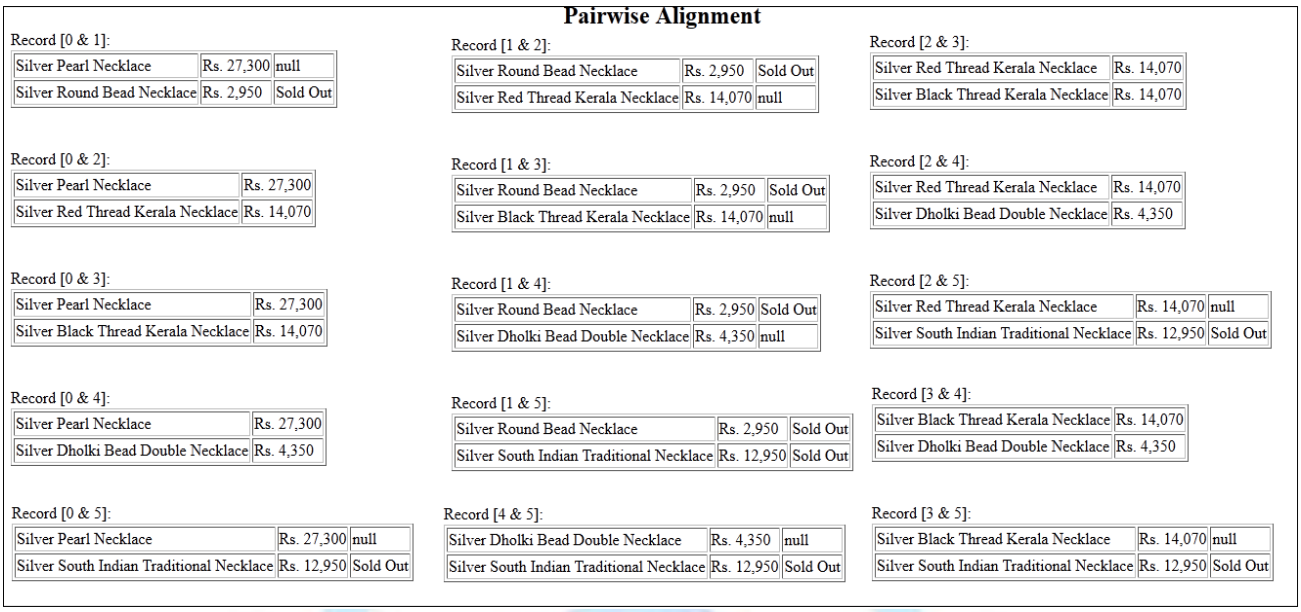
Record:2
Silver Red Thread Kerala Necklace
Rs. 14,070

Record:3
Silver Black Thread Kerala Necklace
Rs. 14,070

Record:4
Silver Dholki Bead Double Necklace
Rs. 4,350

Record:5
Silver South Indian Traditional Necklace
Rs. 12,950
Sold Out

Fig. 7 HTML file showing Extracted QRRs



Pairwise Alignment

Record [0 & 1]:

Silver Pearl Necklace	Rs. 27,300	null
Silver Round Bead Necklace	Rs. 2,950	Sold Out

Record [1 & 2]:

Silver Round Bead Necklace	Rs. 2,950	Sold Out
Silver Red Thread Kerala Necklace	Rs. 14,070	null

Record [2 & 3]:

Silver Red Thread Kerala Necklace	Rs. 14,070
Silver Black Thread Kerala Necklace	Rs. 14,070

Record [0 & 2]:

Silver Pearl Necklace	Rs. 27,300
Silver Red Thread Kerala Necklace	Rs. 14,070

Record [1 & 3]:

Silver Round Bead Necklace	Rs. 2,950	Sold Out
Silver Black Thread Kerala Necklace	Rs. 14,070	null

Record [2 & 4]:

Silver Red Thread Kerala Necklace	Rs. 14,070
Silver Dholki Bead Double Necklace	Rs. 4,350

Record [0 & 3]:

Silver Pearl Necklace	Rs. 27,300
Silver Black Thread Kerala Necklace	Rs. 14,070

Record [1 & 4]:

Silver Round Bead Necklace	Rs. 2,950	Sold Out
Silver Dholki Bead Double Necklace	Rs. 4,350	null

Record [2 & 5]:

Silver Red Thread Kerala Necklace	Rs. 14,070	null
Silver South Indian Traditional Necklace	Rs. 12,950	Sold Out

Record [0 & 4]:

Silver Pearl Necklace	Rs. 27,300
Silver Dholki Bead Double Necklace	Rs. 4,350

Record [1 & 5]:

Silver Round Bead Necklace	Rs. 2,950	Sold Out
Silver South Indian Traditional Necklace	Rs. 12,950	Sold Out

Record [3 & 4]:

Silver Black Thread Kerala Necklace	Rs. 14,070
Silver Dholki Bead Double Necklace	Rs. 4,350

Record [0 & 5]:

Silver Pearl Necklace	Rs. 27,300	null
Silver South Indian Traditional Necklace	Rs. 12,950	Sold Out

Record [4 & 5]:

Silver Dholki Bead Double Necklace	Rs. 4,350	null
Silver South Indian Traditional Necklace	Rs. 12,950	Sold Out

Record [3 & 5]:

Silver Black Thread Kerala Necklace	Rs. 14,070	null
Silver South Indian Traditional Necklace	Rs. 12,950	Sold Out

Fig. 8 HTML file showing Pairwise Aligned QRRs

Holistic Alignment

Silver Pearl Necklace	Rs. 27,300	
Silver Round Bead Necklace	Rs. 2,950	Sold Out
Silver Red Thread Kerala Necklace	Rs. 14,070	
Silver Black Thread Kerala Necklace	Rs. 14,070	
Silver Dholki Bead Double Necklace	Rs. 4,350	
Silver South Indian Traditional Necklace	Rs. 12,950	Sold Out

Fig. 9 HTML file showing Holistically Aligned QRRs

PERFORMANCE EVALUATION

The system was successfully tested on Data set 1 (TBDW version 1.02) and Data set 2 (www.tribebyamarpali.com). Two sets of evaluation metrics are used to evaluate performance. The first set is record level precision and recall metrics can be defined as,

$$P_r = N_c / N_e \text{ and } R_r = N_c / N_r$$

Where N_c is the number of correctly extracted and aligned QRRs, N_e is the number of extracted QRRs, and N_r is the actual number of QRRs in the query result pages.

The number of QRRs in different query result pages varies from a few to hundreds. Consequently, pages with many QRRs will dominate the record-level metrics. To deal with this problem, we also use a page-level metric, namely, page-level precision defined as,

$$P_p = N_p / N_a$$

Where N_p is the number of correctly extracted pages, which means that all the QRRs in the pages are correctly extracted and aligned, and N_a is the number of all the pages from which QRRs are extracted. The page-level recall is always equal to the page-level precision because we assume that each of the input pages contains at least two QRRs and the data extraction is performed on all input pages.

Table1 shows the experimental results for Automatic WDSA. Table 1 shows that Automatic WDSA can extract and align QRRs very effectively, with both record level precision and recall around 98 percent on Dataset 1 and 100 percent on Dataset 2, and page level precision 95 and 100 percent on Dataset 1 and 2 respectively.

ISSN 2277-3061

Table 1 Data Extraction and Alignment Performance Achieved on Dataset 1 and Dataset 2

Dataset	TBDW 1.02V		www.india.isharya.com		www.jabong.com		www.tribebyamarpali.com	
No of pages	65		200		300		30	
Actual QRRs	#1665		#6000		#10000		#1100	
	Extraction	Alignment	Extraction	Alignment	Extraction	Alignment	Extraction	Alignment
#Correct	#1665	#1634	#6000	#6000	#10000	#10000	#1100	#1100
#Wrong	Nil	#31	Nil	Nil	Nil	Nil	Nil	Nil
Record Level Precision	100%	98.13%	100%	100%	100%	100%	100%	100%
Record Level Recall	100%	98.13%	100%	100%	100%	100%	100%	100%
Page Level Precision and Recall	100%	95.38%	100%	100%	100%	100%	100%	100%

We have also compared the performance of Automatic WDSA with existing most widely used famous tools like Visual web ripper, Web content extractor, Outwit Hub, Screen Scraper etc. During comparison we observed that Manual labeling or extraction patterns of desired data is to be specified for working of existing tools while Automatic WDSA does not require any user involvement which made it fully automatic. Fig. 10 shows the summarization of data extraction tools which gives the comparison with respect to different characteristics exhibited by these tools.

Features \ Tools	Automatic WDSA	Visual Web Ripper	Web content extractor	OutWit Hub	Screen Scraper
Need to specify extraction patterns	✗	✓	✓	✗	✓
Need to specify Regular Expression patterns	✗	✗	✗	✓	✗
Manual labeling required	✗	✓	✓	✗	✗
Need of coding	✗	✗	✗	✗	✓
Configuration require	✗	✗	✗	✗	✓
Export data as html file	✓	✗	✓	✓	✗

Fig. 10 Web Data Extraction Tool's Summarization

Further response time of Automatic WDSA is compared with that of Visual web ripper and Web content extractor. The fig. 11 shows graph which reflects that the response time of Automatic WDSA is much less than others. The fig 12 shows that Automatic WDSA achieved better performance than Visual web ripper and Web content extractor respectively by reducing the response time required after submitting the query result page.

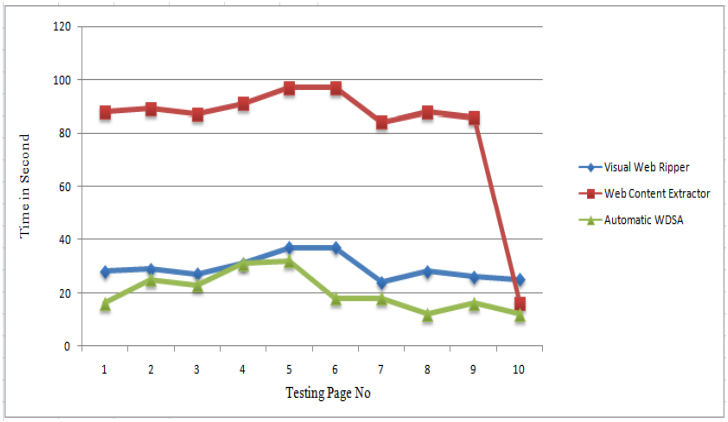


Fig. 11 Response Time Graph

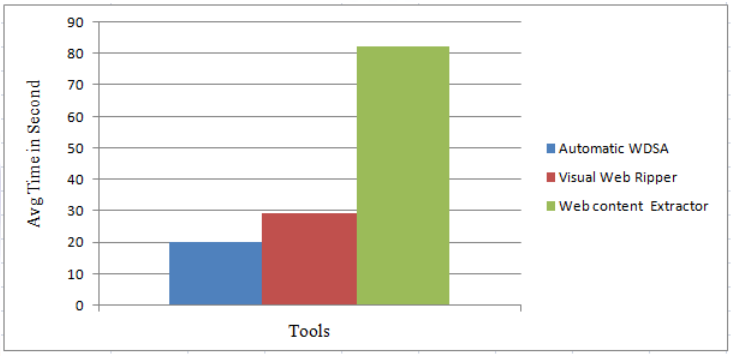


Fig. 12 Average Response Time Graph

CONCLUSION

In this paper, we dealt with solving the problem of data extraction from web pages and to make them usable by representing those in proper format which can be used in applications like value added services, data integration, comparison shopping etc. Particularly, main aim is to find a way to automatically extract query relevant data and convert them in a standard format like tables. Automatic Web Data Scraper and Aligner (WDSA) works in two steps: Data scraping and Data Alignment. Data scraping performs automatic data extraction to get the QRRs from input query result page. The data values of QRRs are aligned by novel step, data alignment in pairwise and holistic fashion by using similarity of both tag and value which differentiates our method from others. Experimental results on Data set 1 and Data set 2 demonstrated the effectiveness of our method.

REFERENCES

[1] B. Liu, R. Grossman, and Y. Zhai, "*Mining Data Records in Web Pages"*, Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 601-606, 2003.

[2] B. Liu and Y. Zhai, "*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.

[3] C.H. Chang and S.C. Lui, "IEPAD: Information Extraction Based on Pattern Discovery," Proc. 10th World Wide Web Conf., pp. 681-688, 2001.

[4] F. H. Lochovsky, W. Su, J. Wang 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.

[5] 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.

[6] Heikki Hyyr, "Practical Methods for Approximate String Matching", Department of Computer Sciences. University of Tampere, Finland.

[7] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. 12th World Wide Web Conf.,pp. 187-196, 2003.

[8] J. Wang and F. Lochovsky, "*Data-Rich Section Extraction from HTML Pages"*, Proc. Third IntŠl Conf. Web Information System Eng., 2002.

[9] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. 12th World Wide Web Conf., pp. 187-196, 2003.

[10] K. Simon and G. Lausen, "*ViPER: Augmenting Automatic Information Extraction with Visual Perceptions"*, Proc. 14th ACM Int'l Conf. Information and Knowledge Management, pp. 381-388, 2005.

[11] M. Alvarez, A. Pan, J. Raposo, F. Bellas, and F. Cacheda, "Using Clustering and Edit Distance Techniques for Automatic Web Data Extraction", Springer-Verlag Berlin Heidelberg, pp. 212–224, 2007.

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

[13] R. Baumgartner, S. Flesca, and G. Gottlob, "*Visual Web Information Extraction with Lixto"* , Proc. 27th Int'l Conf. Very Large Data Bases, pp. 119-128, 2001.

[14] S. swami and P. Vidap," *Web Scraping Framework Based on Combining Tag and Value Similarity",* IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 6,November 2013

[15] V. Crescenzi, G. Mecca, and P. Merialdo, "Roadrunner: Towards Automatic Data Extraction from Large Web Sites," Proc. 27th Int'l Conf. Very Large Data Bases, pp. 109-118, 2001.

[16] W. Cohen, M. Hurst, and L. Jensen, "A Flexible Learning System for Wrapping Tables and Lists in HTML Documents," Proc. 11th World Wide Web Conf., pp. 232-241, 2002.

[17] W. Han, L. Liu and C. Pu, "XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources," Proc. 16th Int'l Conf. Data Eng., pp. 611-621, 2000.