Data Extraction and Alignment Techniques for effective mining of Query Result Pages

Ramanan.M , Lourdu Michael Antony.K, Boopathy.M

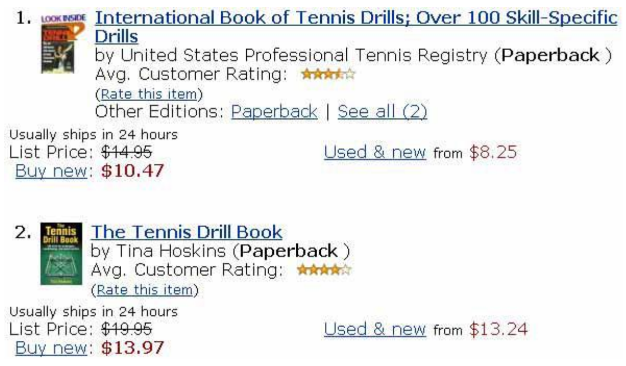
[*] Assistant Professor, Department of Information Technology, Park College of Engineering & Technology, Coimbatore

***Abstract*-** Deep web comprises the online database, which generates web pages dynamically in response to a user query. The extracted data is used for many applications like meta querying and comparison shopping. For these applications the data is embedded in HTML pages. The automatic data extraction is necessary for utilizing these data. Only when the data are extracted and organized in a structured manner, such as tables, they can be compared and aggregated. An effective data extraction and alignment approach is used which utilizes both tag and value similarity in a web page. This method automatically extracts data from query result pages by identifying and segmenting the query result records (QRRs) in a query result page. The segmented QRRs are arranged in a table accordingly with data values from the same attribute. They are put into the same column using pairwise and global arrangement. This new technique also handles non continuous data regions that may exist in a web page due to some advertisements, additional information. Experimental results show that our system can achieve high accuracy in extracting and aligning structured objects inside complex web pages.

***Index Terms*-** Data extraction, automatic wrapper generation, Data record arrangement, information integration.

I. INTRODUCTION

Web content mining is extraction and integration of useful information and knowledge from Web page contents. Web content mining process involves in automatic content extraction from web pages, integration of the information, knowledge synthesis, noise detection and segmentation. Due to the heterogeneity and lack of structure of Web data, mining is a challenging task. This paper focuses on the problem of automatically extracting data records that are encoded in the query result pages generated by online databases. The goal of data extraction from web database is to remove irrelevant information from the generated query result page and extract the query result records from the web page, and align the extracted QRRs into a table such that the data values belonging to the same attribute are placed into the same table column.


1. International Book of Tennis Drills; Over 100 Skill-Specific Drills
by United States Professional Tennis Registry (Paperback)
Avg. Customer Rating:
(Rate this item)
Other Editions: Paperback | See all (2)
Usually ships in 24 hours
List Price: ~~$14.95~~ Used & new from $8.25
Buy new: $10.47

2. The Tennis Drill Book
by Tina Hoskins (Paperback)
Avg. Customer Rating:
(Rate this item)
Usually ships in 24 hours
List Price: ~~$19.95~~ Used & new from $13.24
Buy new: $13.97

Fig. 1. An example query result page for the query—Book: Tennis Drills.

For example, Figure 1 gives some example data records on the Web. Figure 1 shows a Web page segment containing of query result records for list of two products (books). The description of each book is a data record.

Our objective in data extraction and alignment process is to
(1) Identify such data records in a page automatically.
(2) Align and extract data items from the data records automatically.

We employ the following two-step method, called Combining Tag and Value Similarity, to extract the QRRs from a query result page p.

1. Data extraction identifies the QRRs in p and involves two sub steps: data region recognition and the actual segmentation step.

2. Record alignment aligns the data values of the QRRs in p into a table so that data values for the same attribute are aligned into the same table column.

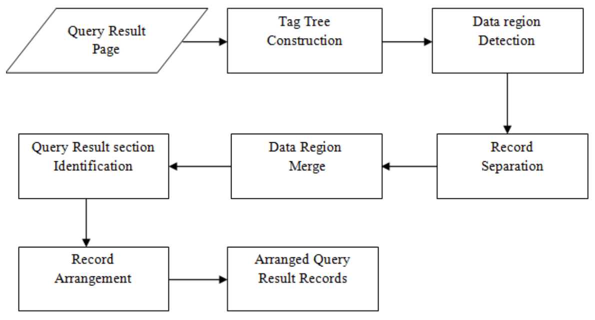


Fig. 2. Data Extraction and Alignment Framework

Compared with existing data extraction methods, our extraction method improves data extraction accuracy in three ways.

1. The method in [2] can find all data regions containing at least two QRRs in a query result page using mining techniques, likewise all other data extraction methods, such as [1] and [6], assume that the QRRs are presented adjacently in only one data region in a page. We examined some websites to determine the extent to which the QRRs in the query result pages are not continuous. We found that the QRRs in most of the websites are non contiguous, indicating that noncontiguous data regions are common in majority of websites. By analyzing this problem, we employ two techniques according to the layout of the QRRs and the auxiliary information in the result page's HTML tag trees (i.e., DOM trees).

2. An efficient method is employed to align the data values in the identified Query Result Records, first pairwise then globally, so that they can be put into a table with the data values belonging to the same attributes are arranged into the same table column.

3. A new nested-structure processing algorithm is used to handle any nested structure in the QRRs after the holistic arrangement. Unlike existing nested-structure processing algorithms that rely on only tag information, this algorithm uses both tag and data value similarity information to improve nested structure processing accuracy.

II. RELATED WORKS

Web database extraction has received much attention from the Database and Information Extraction research areas in recent years due to the volume and quality of deep web data. Wrapper induction involves generalizing from a set of examples of a resource's pages, each annotated with the text fragments to be extracted. The most interesting challenges are that existing wrapper induction systems ignores the fact that the formatting conventions on which wrappers rely can change unexpectedly. The implicit strategy is to learn a new wrapper from scratch, rather than repair the broken wrapper.

World Wide Web Wrapper Factory (W4F) [5], a toolkit for the generation of wrappers for Web sources, that offers an expressive language to specify the extraction of complex structures from HTML pages and a declarative mapping to various data formats like XML and utilize some visual tools to make the engineering of wrappers faster and easier. All these Semi Automated approaches require manual efforts like labeling some sample pages. It is labor-intensive and time consuming.

In order to improve the efficiency most recent researches focus on automatic approaches. Some representative automatic approaches are ViNT [4], DeLa [1], MDR [2], ViPER [3]. These methods rely entirely on the tag structure in the query result pages. ViNTs [6] which focuses on the issue of how to extract search result records (SRRs) from dynamically generated result pages returned by search engines in response to submitted queries. DeLA which models the structured data contained in template generated web pages as string instances encoded in HTML tags. MDR currently finds all data records formed by table and form related tags. A large majority of web data records are formed by them. The problem with this approach is that the computation is prohibitive because a data record can start from anywhere and end anywhere. In ViPER, assumption made is that a Web page contains at least two multiple spatially consecutive data records building a data region which exhibits some kind of structural and visible similarity.

By using similarity of tag and value which has better performance than ViNTs and DeLa in both non nested and nested pages. For non nested pages, each of the three methods has performance greater than 95 percent for the record-level precision and 94 percent for the record level recall. For the nested pages, CTVS and While the performance of ViNTs drops, DeLa still attain good record-level precision and recall. It is observed that the nested pages are usually more complex than the non nested pages. This more complex structure causes the performance of ViNTs to degenerate.

Each method's performance is greater than 95 percent on the record-level precision and 93 percent on the record-level recall for pages with continuous data. For noncontiguous pages, CTVS maintains a fairly high page-level precision when compared with ViNTs and DeLa. CTVS high page-level precision for noncontiguous pages is due to its data region identification and merging algorithms, which are designed to deal with noncontiguous QRRs. Inversely, if the actual data region is split into two or more parts, DeLa only report the QRRs from the largest subpart and ignore others.

However, unlike ViNTs, similarity of tag and value cannot handle no-result pages, since it assumes there are at least two QRRs in the page to be extracted. All the preceding works make use of only the information in the query result pages to perform the data extraction.

III. DATA EXTRACTION

Given a query result page, the Tag Tree Construction module first constructs a tag tree for the page rooted in the <HTML> tag. Data Region Detection module identifies all possible data regions of dynamically generated data starting from the root node, traversing through the web page. The Data Record Segmentation module segments the identified data regions into data records based on the data region's tag patterns. With the segmented data records, the data regions containing similar records are merged in the Data Region Merge module. Finally, one of the merged data regions is selected as the one that contains the QRRs the Query Result Section Recognition module.

A. Tag Tree Construction

Tag tree construction finds all data records formed by table and form related tags, i.e. <html>, <form>, <table> and so on. Major Web data records are formed by them. Tag tree construction is based on two observations:

1. A group of data records that contains descriptions of a set of objects are presented in a varied region of a page and are formatted by HTML tags.

2. The nested structure of HTML tags in a Web page naturally forms a tag tree and data may not present in child node under same parent tag. A new algorithm to employed to process such nested structures.

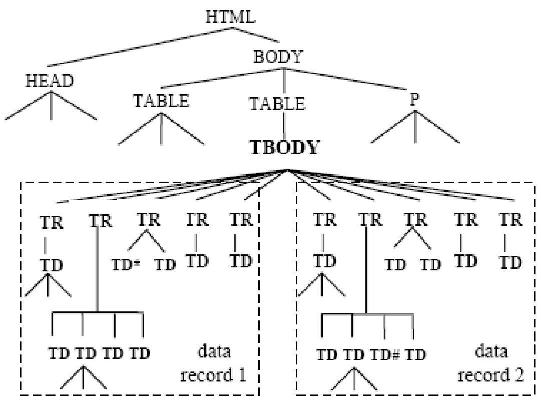


Fig. 3. Tag tree of the page in Fig. 1

B. Data Region Detection

We propose a new method to handle noncontiguous data regions so that it can be applied to more web databases. Under the assumption that there are at least two QRRs in a query result page, the data region detection algorithm discovers data regions in a top down manner. Starting from the root tag of the query result page tag tree, the data region detection algorithm is applied to a node n and recursively to its children n_i, i ¼, 1 ……..m as follows:

1. Compute the similarity sim_{ij} of each pair of nodes ni and n_j where i,j = 1...m and i ≠ j, using the node similarity calculation method presented later in this section. The data region recognition algorithm is recursively applied to the children of n_i only if it does not have any similar siblings. The similar nodes recognized with the same parent forms a data region. Likewise other similar data regions may be identified in this step.

2. Segment the data region into data records using the record segmentation algorithm is described later. Suppose that the tag tree has n internal nodes and a node has a maximum of m children and a maximum tag string length of l.

To calculate the node similarity, the method used in ViPER [3] and DEPTA [6] is adopted. Two nodes n1 and n2 are similar if their similarity, sim12, is larger than or equal to a threshold T_{nsim}. For n1 and n2, their corresponding tag strings s1 and s2 are used to evaluate their similarity. To overcome the problem caused by optional attributes and repetitive subparts, Simon and Lausen [3] suggest that the similarity threshold T_{nsim} be set to 0.6.

The time complexity of the data region detection algorithm is $O(nm^2l^2)$.

C. Record Separation

In the record separation module we will see for Tandem repeats. It refers to the repeated tag substrings which are directly adjacent to each other. If only a single tandem repeat is identified in a data region, then each repeated instance inside the tandem repeat corresponds to a record. If multiple tandem repeats are found in a data region, we need to select one to denote the record. The following two scenarios are considered to select a tandem repeat.

1. If there is auxiliary information is present between record instances then the tandem repeat will stop at auxiliary information.

2. When compared with any visual gap within a record the visual gap between two records in a data region is usually larger. Hence, the tandem repeat that satisfies this constraint is selected.

3. We select the tandem repeat that starts the data region If the above two heuristics cannot be used.

Nevertheless, our data region detection algorithm can still identify the data region that contains the noncontiguous QRRs.

D. Data Region Merge

The data region detection step identifies several data regions in a query result page. But the actual data records may span across several data regions. Before identifying all the Query Result Records, it is to be determined that whether any of the data regions should be merged. Given any two data regions, we treat them as similar if the segmented records they contain are similar. The similarity between two data regions is calculated as the average record similarity.

Two data regions can be merged into a merged data region if the records in the two data regions have an average similarity ranging near 0.6, which is a threshold used to judge whether two records are similar in [3].

E. Query Result Section Detection

After performing the data region merge step, still multiple data regions may exists in a query result page. Three heuristics are used to detect this data region, called the query result section.

1. The query result section usually occupies a large space in the query result page [7]. For every data region (d), area weight is calculated as d's area divided by the largest area of all identified data regions, is assigned for d.

2. The query result section is usually located at the center of the query result page [7]. For each data region d, a distance between its center and the center of the page is center distance, and the smallest center distance among all identified regions is center distance weight which is divided by d's center distance, is assigned for d.

3. Each QRR usually contains more raw data strings than the raw data strings in other sections. For each data region d, a value weight is calculated as the average number of raw data strings in the records of d divided by the largest average number of data values in all identified regions, is assigned for data region d.

The above three weights are summed and the query result section is selected from the data region that having the largest summed weight. Records in this data region are assumed to be Query Result Records.

A limitation of this approach is that if a query result page as more than one data region that contains query result records and the records in the different data regions are not similar, then only one of the data regions is selected and others will be discarded. This case is observed in least sites among the surveyed websites.

IV. QRR ARRANGEMENT

Query Result Record arrangement is performed by a three-step data arrangement method that combines tag and value similarity.

1. In Pairwise QRR arrangement the data values are aligned in a pair of QRRs to provide the evidence for how the data values should be aligned among all QRRs.

2. Global Data Record arrangement that arranges the data values in all the QRRs.

3. In Nested structure processing, the nested structures that exist in the QRRs are identified.

A. Pairwise QRR Arrangement

The pairwise QRR arrangement algorithm is based on the observation that the data values belonging to the same attribute usually have the same data type and may contain similar strings, since the QRRs are for the same query. Given two QRRs r1={f_{11}, …, f_{1m}} where f_{1i} refers to the ith data value of r1, and r2={f_{21}, …, f_{2n}} , we first calculate the data value similarity, s_{ij}, between every pair of data values in r1 and r2.

During the pairwise arrangement, we require that the data value arrangements must satisfy the following three constraints:

1. Same record path constraint. The record path of a data value f comprises the tag from the root of the record to the node that contains f in the tag tree of the query result page. Every pair of matched values should have same tag path. Hence, if f_{1i} has a different tag path with f_{2j}, then s_{ij} is assigned a small negative value to prevent the pair of values from being aligned.

2. Unique constraint. Each data value can be aligned to at most one data value from the other QRR.

3. No cross arrangement constraint. If f_{1i} is matched to f_{2j}, then there should be no data value arrangement between f_{1k} and f_{2l} such that $k < i$ and $l > j$ or $k > i$ and $l < j$.

B. Global Data Record Arrangement

The QRRs are arranged globally among them to construct a table in which all data values of the same attribute are aligned in the same table column. If we view each data value In the QRRs as a vertex and each pairwise arrangement between two data values as an edge, the pairwise arrangement set can be viewed as an undirected graph. Thus, finding connected components in an undirected graph is our global data arrangement problem. Each connected component of the graph represents a table column inside which the connected data values from different records are aligned vertically. There are two application constraints that are specific to our global data arrangement problem.

1. Vertices from the same record are not allowed to be included in the same connected component as they are considered to come from two different attributes of the record. A path must exist between the two vertices, if two are from the same record breach this constraint, which we call a breach path.

2. Connected components are not allowed to intersect each other. If C1 and C2 are two connected components, then vertices in C1 should be either all on the left side of C2 or all on the right side of C2,and vice versa (i.e., no edge in C1 cuts across C2, and no edge in C2 cuts across C1).

```
Function GlobalAlign(G,T)
Input : pairwise aligment graph G
Output: globally aligned table T
1. $C_{left} = C_{right} = Ø$
2. if C not empty then
3. find component C[i] in C having maximum edges
4. insert globally aligned column in T by
   aligning all vertices in C[i] in the column
5. for each component C[k] in C, K•i
6. if C[k] intersect with C[i]
   then
7. $C[k]_{left}$ = vertices of C[k] left to C[i]
8. $C[k]_{right}$= vertices of C[k] right to C[i]
9.  remove edges connecting $C[k]_{left}$ and $C[k]_{right}$
10. $C_{left} = C_{left} + C[k]_{left}$
11. $C_{right} = C_{right} + C[k]_{right}$
12. else if C[k] is to left of C[i] then
13. $C_{left} = C_{left} + C[k]$
14. else
15. $C_{right} = C_{right} + C[k]$
16. GlobalAlign($C_{left}$,T)
17. GlobalAlign($C_{right}$,T)
```
Fig. 4. Global Data Record Alignment Algorithm

C. Nested Structure processing

Global data value arrangement constrains a data value in a Query Result Record to be aligned to at most one data value from another Query Result Record. A QRR containing a nested structure having multi valued attribute, then some of the values may not be aligned to any other values. For this the nested structure processing identifies the data values of a QRR that are generated by nested structures. The nested structure processing method in similarity of Tag and value has the following advantages.

1. CTVS processes the nested structures after the data records are aligned rather than before as is the case in DeLa[1] and NET[8]. Processing the nested structure before the records are aligned makes them vulnerable to optional attributes since the optional attributes make the tag structure irregular. CTVS avoids this problem.

2. The data value similarity information effectively prevents a flat structure from being identified as a nested structure in CTVS. As it shares similar tag structures, a structure with several columns having the same tag structure, may be identified as a nested structure mistakenly in DeLa. Such wrongly identified flat structure can have serious effects. DeLa groups all the values into one parent and then aligns them to other records, making the arrangement much more difficult.

```
Procedure nest_processing(QRRs,T,global_align)
1. C□ Ø
2. for each QRR with record root t
3. nest_column _identify(t,T,global_align,C)
4. for each column pattern $c_p$ in C do
5. create a new row for each repeated subpart
Procedure nest_column_identify(t,T,global_align,C)
1. if( t contains more than one data value) then
2. for each child $t_i$ of t do
3. nest_column_identify($t_i$,T,global_align,C)
4. for each repetition p of any consecutive maximum
repetitive tag found in t's children
5. $c_p$ = data columns for p in global_align
6. if $c_p \notin$ C data and nested ($c_p$,Snest) then
7. add_nested_column($c_p$,C)
Function boolean_nested($c_p$,Snest)
1. $sim_{intra} \leftarrow$ intra- column similarity within $c_p$
2. $sim_{inter} \leftarrow$inter- column similarity within $c_p$
3. if($sim_{intra}$/ $sim_{inter}$> /Snest) then
```

```
4. return true
5. else return false
```
Fig 5: Nested structure processing algorithm

V. EXPERIMENTS

We now present the experimental results for CTVS over five data sets and compare CTVS with ViNTs [4], DeLa [1], and ViPER [3]. We choose ViNTs and DeLa to compare with CTVS. In the Data Extraction performed with CTVS, the extracted records precision and accuracy is higher when compared with DeLa and ViPER methods. While using mentioned two data sets, CTVS consistently has the best performance on both data sets. ViNTs is for QRRs with simple structures used document databases, while it degenerates quickly for QRRs with complex structures but CTVS outperforms with complex structures. CTVS is implemented using C#.

A. Data Sets

Data set 1 (PROFUSION) is obtained from ViNT's test bed which has 100 websites of government, education and healthcare domains. Data set 2 is the TestBed for information extraction from Deep web.

VI. CONCLUSION

We presented a new data extraction method to automatically extract QRRs from a query result page. In our method, we first identified and segmented the query result records. Our extraction method is efficient in handling non contiguous QRRs from the existing methods. First, it requires at least two query result record in a result page. Second, if any optional attribute appears as the start node in a data region, then that data region will be treated as auxiliary information.

Although our algorithm has been shown to be an accurate data extraction method, it has some limitations. If a query result page has more than one data region that contains result records and the records in the different data regions are not similar to each other, then it will select only one of those data regions and discard the others. Our algorithm mainly depends on tag structures to discover data values. Therefore, it does not handle the case where multiple data values from more than one attribute are clustered inside one leaf node of the tag tree, also the case where one data value of a single attribute spans multiple leaf nodes. This can be considered as future enhancement.

REFERENCES

[1] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. 12th World Wide Web Conf., pp. 187-196, 2003
[2] 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.
[3] 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.
[4] 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.
[5] A. Sahuguet and F. Azavant, "Building Intelligent Web Applications Using Lightweight Wrappers," Data and Knowledge Eng.,vol. 36, no. 3, pp. 283, 2001.
[6] 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.
[7] 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.
[8] B. Liu and Y. Zhai, "NET - A System for ExtractingWeb Data from Flat and Nested Data Records," Proc. Sixth Int'l Conf. Web Information Systems Eng., pp. 487-495, 2005.
[9] H. Snoussi, L. Magnin, and J.-Y. Nie, "Heterogeneous Web Data Extraction Using Ontologies," Proc. Fifth Int'l Conf. Agent -Oriented Information Systems, pp. 99 -110, 2001.
[10] A. Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 337-348, 2003.
[11] R. Baeza-Yates, "Algorithms for String Matching: A Survey," ACM SIGIR Forum, vol. 23, nos. 3/4, pp. 34-58, 1989.
[12] 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.
[13] V. Crescenzi and G. Mecca, "Grammars Have Exceptions," Information Systems, vol. 23, no. 8, pp. 539-565, 1998.
[14] G.O. Arocena and A.O. Mendelzon, "WebOQL: Restructuring Documents, Databases, and Webs," Proc. Int'l Conf. Data Eng. (ICDE), pp. 24-33, 1998.

AUTHORS

First Author – Ramanan.M, Assistant Professor, Department of Information Technology, Park College of Engineering & Technology, Coimbatore, Email: pmramanan@gmail.com
Second Author – Lourdu Michael Antony.K, Assistant Professor, Department of Information Technology, Park College of Engineering & Technology, Coimbatore, Email: lourdumike@gmail.com
Third Author – Boopathy.M, Assistant Professor, Department of Information Technology, Park College of Engineering & Technology, Coimbatore, Email: boopathymtec@gmail.com