International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

# Visual Resemblance Based Content Descent for Multiset Query Records using Novel Segmentation Algorithm

S. Ishwarya[1], S. Grace Mary[2]

*Department of Computer Science and Engineering, Shivani Engineering College, Trichy, India.*

iswarya7390@gmail.com[1], gracemarychristopher@gmail.com[2]

***Abstract***

**Online data request and respond to a user query with result records are programmed in HTML files. Extracting information from the unstructured bases has matured into a significant technical challenge whereas generally, data extraction had to deal with changes in physical hardware plans, the majority of current data mining deals with extracting data from the un-structured data sources, and from dissimilar software plans. In this paper, we focus on the problem of automatically extracting data records that are encoded in the query result pages generated by web data records. We propose an unusual data extraction scheme called improved combined tag and value similarity (R-SEGMENT algorithm) approach. R-SEGMENT algorithm frequently extracts the query outcome pages by first classifying and metameric the QRR in the query consequence pages and then bring into line the metameric QRRs into a table, in which the data values from the identical attribute are put into the same column. Experimental results show that our system can achieve high accuracy in distilling and aligning regularly structured objects inside complex web pages.**

***Keywords*—Data extraction, automatic wrapper generation, data record arrangement, information integration.**

## I. INTRODUCTION

Online web databases comprises the deepweb [4] and [7]. Compared with webpages in the surface web, which can be accessed by a unique URL, pages in the deep web are dynamically generated in response to a user query submitted through the query interface of a web database. Upon receiving a user's query, a web database returns the relevant data, either structured or semistructured, encoded in HTML pages. For automatic data extraction , data should be managed and organized in a structured manner, such as tables, can they be compared and aggregated. Hence, accurate data extraction is vital for these applications to perform correctly. This paper focuses on the problem of automatically extracting data records that are encoded in the query result pages generated by web databases. In general, a query result page contains not only the actual data, but also other information, suchas navigational

panels, vertisements, comments, information about hosting sites, and so on. The goal of web database data extraction is to remove any irrelevant information from the query result page, extract the query result records.

For example, Fig. 1 shows a query result page fragment containing two QRRs for DKNY products in which the second QRR contains a nested structure with the template "Size: <size>, Color: <color> <price>" and the label "Top Rated" as well as the vertical line between the two QRRs represent auxiliary information.

Table 1 show the aligned table for the two QRRs in Fig. 1 where the third row is generated from the nested information for the second QRR in Fig. 1.

| | |
|---|---|
| DKYN Iridescent Sheer<br>Bed skirts<br>Size:queen , color:Barley<br>$59.99 | DKYNpure hand kerchief<br>flat sheets<br>Size:queen, color:Red, $29<br>Size:king, color:Green, $39 |

**Fig. 1. An example query result page for the query—Brand: DKNY.**

We employ the following two-step method, called Improced version of Combining Tag and Value Similarity (R-SEGMENT algorithm), to extract the QRRs from a query result page p.

1. Record extraction identifies the QRRs in p and involves two substeps: data region recognition and the actual segmentation step.
2. Record arrangement 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.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

Compared with existing data extraction methods, R-SEGMENT algorithm improves data extraction accuracy in three ways.

1. The method in [1] can find all data regions containing at least two QRRs in a query result page using data mining techniques, almost all other data extraction methods, such as [2] and [3], assume that the QRRs are presented contiguously in only one data region in a page. We examined 100 websites to determine the extent to which the QRRs in the query result pages are noncontiguous. We found that the QRRs in 26 out of the 100 websites are noncontiguous, which indicates that noncontiguous data regions are quite common. To address this problem, we employ two techniques according to the layout ofthe QRRs and the auxiliary information in the result page's HTML tag trees (i.e., DOM trees).
2. A novel method is proposed to align the data values in the identified QRRs, first pairwise then holistically, so that they can be put into a table with the data values belonging to the same attribute arranged into the same table column.
3. A new nested-structure processing algorithm is proposed to handle any nested structure in the QRRs after the holistic arrangement. Unlike existing nested-structure processing algorithms that rely on only tag information, R-SEGMENT algorithm uses both tag and data value similarity information to improve nested structure processing accuracy.

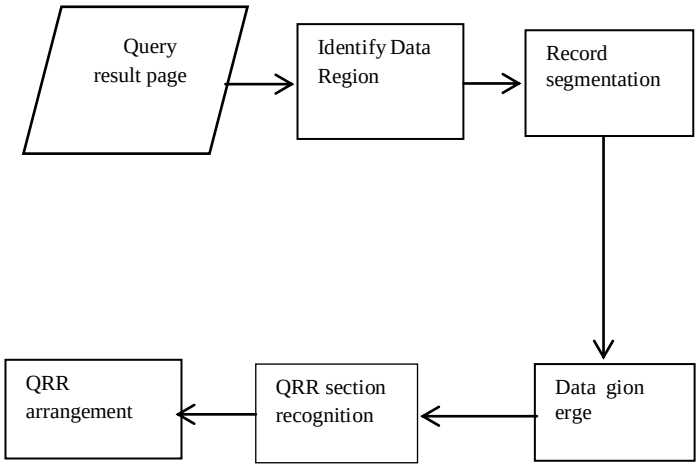


**Fig2: QRR extraction Framework**

## II. QRR 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 Recognition module identifies all possible data regions, which usually contain dynamically generated data, top down starting from the root node. The Record Segmentation module then segments the identified data regions into data records according to the tag patterns in the data regions. Given the segmented data records, the Data Region Merge module merges the data regions containing similar records. Finally, the Query Result Section Recognition module selects one of the merged data regions as the one that contains the QRRs.

### *A. Data Region Recognition*

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 recognition algorithm discovers data regions in a top-down manner. Starting from the root of the query result page tag tree, the data region recognition algorithm is applied to a node n and recursively to its children ni, i ¼ 1 ……..m as follows:

1. Compute the similarity simij of each pair of nodes ni and nj; i; j ¼ 1 . . .m and i 6¼ j, using the node similarity calculation method presented later in this section. The data region recognition algorithm is recursively applied to the children of ni only if it does not have any similar siblings. For example, the algorithm has to be applied to the children of node 5 since it does not have any similar sibling node. The recognized similar nodes with the same parent form a data region. Multiple 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. The time complexity of the data region Recognition algorithm is O(nm2l2).

### *B. Data Segmentation*

If only one tandem repeat is found in a data region, we assume that each repeated instance inside the tandem repeat corresponds to a record, such as nodes 12 and 13 in Region 2. If multiple tandem repeats are found in a data region, such as in Region 1 , we need to select one to denote the record.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

1. If there is auxiliary information, which corresponds to nodes between record instances, within a data region, the tandem repeat that stops at the auxiliary information is the correct tandem repeat since auxiliary information usually is not inserted into the middle of a record.

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

3. If the preceding two heuristics cannot be used, we select the tandem repeat that starts the data region.

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

C. *Data Region Merge*

The data region recognition step may identify several data regions in a query result page. Moreover, the actual data records may span several data regions. The data region recognition step may identify several data regions in a query result page. Moreover, the actual data records may span several data regions. 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 greater or equal to 0.6, which is a threshold used to judge whether two records are similar in [24].

D. *Query Result Section Recognition*

Even after performing the data region merge step, there may still be multiple data regions in a query result page. Three heuristics are used to identify this data region, called the query result section.

1. The query result section usually occupies a large space in the query result page [3]. For each data region d, an area weight, which 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 [3]. For each data region d, a center distance is calculated between its center and the center of the page, and a center distance weight, which is calculated as the smallest center distance among all identified regions 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, which 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 d.

A limitation of this approach is that if a query result page as mo re than one data region that contains query result records and the records in the different data regions are not similar to each other, then we will select only one of the data regions and discard the others. This case was observed in 2 out of the 100 websites surveyed.

## III. QRR Arrangement

QRR arrangement is performed by a novel three-step data arrangement method that combines tag and value similarity.

1. Pairwise QRR arrangement aligns the data values in a pair of QRRs to provide the evidence for how the data values should be aligned among QRRs.
2. Holistic arrangement aligns the data values in all the QRRs.
3. Nested structure processing identifies the nested structures that exist in the QRRs.

*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, especially since the QRRs are for the same query.

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. Each pair of matched values should have the same tag path. Hence, if f1i has a different tag path with f2j, then sij 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 f1i is matched to f2j, then there should be no data value arrangement between f1k and f2l such that $k < i$ and $l > j$ or $k > i$ and $l < j$.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

*B. Holistic Arrangement*

Arrangement globally among all qrrs to construct a table in which all data values of the same attribute are aligned in the same table column. Intuitively, 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, our holistic arrangement problem is equivalent to that of finding connected components6 in an undirected graph. 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 holistic 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. If two vertices from the same record breach this constraint, a path must exist between the two, 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 Traverse(G)*

*Input:* pairwise arrangement graph G
*Output:* a list C in which each element is connected component in G with its corresponding breach flag.

1. **for** each vertex u € V(G)
2. color[u]:=WHITE // unvisited
3. i=0
4. **for** each vertex u€V(G)
5. **if** color[u]=WHITE **then**
6. i=i+1
7. C[i].vertices = {u} //start a new component
8. C[i].flag=**Visit**(u,C[i])
9. **for** each component C[i] in C
10. **if** C[i].flag=true **then**
11. **BreakBreachPath**(C[i])
12. update list C
13. **return** C

**Fig 3: Holistic Arrangement Algorithm**

*C. Nested Structure Processing*

Holistic data value arrangement constrains a data value in a QRR to be aligned to at most one data value from another QRR.

If a QRR contains a nested structure such that an attribute has multiple values, then some of the values may not be aligned to any other values. Therefore, 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 and NET. Processing the nested structure before the records are aligned makes them vulnerable to optional attributes since the optional attributes make the tag structure irregular. This problem does not occur in CTVS.

2. In CTVS the data value similarity information effectively prevents a flat structure from being identified as a nested structure. Because it shares similar tag structures, a flat structure with several columns having the same tag structure, might be mistakenly identified as a nested structure in DeLa and NET. Incorrectly identifying a flat structure as a nested one can have serious consequences. DeLa condenses all the values into one parent value and then aligns them to other records, which makes the arrangement much more difficult. If NET incorrectly identifies a plain structure as a nested structure, it will create a new row in the table for each data value.

**Procedure nest_processing(**QRRs,T,holistic_align)
1. C← null
2. **for** each QRR with record root t
3. nest_column _identify(t,T,holistic_align,C)
4. **for** each column pattern Cp in C **do**
5. create a new row for each repeated subpart

**Procedure nest_column_identify**(t,T,holistic_align,C)
1. **if**( t contains more than on e data value) **then**
2. **for** each child $t_i$ of t **do**
3. nest_column_identify(t,T,holistic_align,C)
4. **for** each repetition p of any consecutive maximum repetitive tag pattern found in t's children
5. Cp= data columns for p in holistic_align
6. **if** Cp€ data and nested (Cp,Snot) **then**
7. add_nested_column(Cp,C)

**Function Boolean_nested()**
1. $sim_{intra}$← intra- column similarity within $c_p$
2. $sim_{inter}$ ← inter- column similarity within $c_p$
3. **if**($sim_{intra}$/ $sim_{inter}$> /$S_{not}$) **then**
4. **return** true
5. **else return** false

**Fig 4: Nested structure processing algorithm**

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

## IV. EXPERIMENTS

We now present the experimental results for CTVS over five data sets and compare CTVS with ViNTs [3], DeLa [9], and ViPER [8]. We choose ViNTs and DeLa to compare with CTVS because both have been shown to perform very accurate data extraction and implementations of both are available to us.

The performance result of ViPER is taken from Simon and Lausen [4] and is limited to what is reported there since we did not have its implementation available to us. While there are other state- of-the-art methods, such as DEPTA, to which we would like to have compared CTVS, their available information lacks some implementation details preventing us from implementing them. CTVS is implemented in JAVA and C++.

### *A. Data Sets*

Five data sets, whose properties are summarized in Table 6, are used to compare the performance of CTVS, ViNTs, and DeLa. Data set 1 (PROFUSION) is obtained from ViNTs', Data set 2 (E-COMM) contains 100 E-commerce deep websites in six popular domains: book, hotel, job, movie, musicRecord, and automobile. Data set 3 is the TestBed for information extraction from Deep web. Data set 4 (AUXI) focuses on webpages. Data set 5 (NESTED) focuses on the query result pages that include nested structure.

## V. RELATED WORK

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 [2], [3], [4], and [5]. As the returned data for a query are embedded in HTML pages, the research has focused on how to extract this data. Earlier work focused on wrapper induction methods, which require human assistance to build a wrapper.

In wrapper induction, extraction rules are derived based on inductive learning. A user labels or marks part or all of the item(s) to extract (the target item(s)) in a set of training pages or a list of data records in a page and the system then learns the wrapper rules from the labeled data and uses them to extract records from new pages. A rule usually contains two patterns, a prefix pattern and a suffix pattern, to denote the beginning and the end, respectively, of the target item.

While wrapper induction has the advantage that no extraneous data are extracted, since the user can label only the items of interest, it requires labor intensive and time-consuming manual labeling of data. Thus, it is not scalable to a large number of web databases. Moreover, an existing wrapper can perform poorly when the format of a query result page changes, which may happen frequently on the web.

Hence, the wrapper induction approach involves two further difficult problems: monitoring changes in a page's format and maintaining a wrapper when a page's format changes. Deriving accurate wrappers based solely on HTML tags is very difficult for the following reasons [3]. One cannot rely on "proper" HTML tag usage since HTML tags are often used in unexpected and unconventional ways. HTML tags convey little semantic information since their main purpose is to facilitate the rendering of data.

Some data may contain embedded tags, which may confuse the wrapper generators making them even less reliable. ViPER [24] 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. While ViPER suffers from poor results for nested structured data, CTVS handles nested structured data efficiently and precisely. ViNTs has several drawbacks. First, if the data records are distributed over multiple data regions only the major data region is reported. Second, 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. Third, 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. In contrast, CTVS requires neither training pages nor a prelearned wrapper for a website.

However, unlike ViNTs, SIMILARITY OF TAG AND VALUE cannot handle no-result pages, since SIMILARITY OF TAG AND VALUE 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.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

There are works that make use of additional information, specifically ontologies, to assist in the data extraction ([1], [7], and [8]).

## VI. CONCLUSIONS

We presented a novel data extraction method, R-SEGMENT algorithm, to automatically extract QRRs from a query result page. The first step identifies and segments the QRRs.

We improve on existing techniques by allowing the QRRs in a data region to be noncontiguous. First, it requires at least two QRRs in the query result page. Second, any optional attribute that appears as the start node in a data region will be treated as auxiliary information

As previously mentioned, 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 R-SEGMENT algorithm will select only one of the data regions and discard the others. Although R- SEGMENT algorithm has been shown to be an accurate data extraction method, it still suffers from some limitations. Similar to other related works, SIMILARITY OF TAG AND VALUE mainly depends on tag structures to discover data values.
Therefore, R-SEGMENT algorithm does not handle the case where multiple data values from more than one attribute are clustered inside one leaf node of the tag tree, as well as the case where one data value of a single attribute spans multiple leaf nodes.This can be considered as future enhancement.

## 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 ] 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.

[3 ] Y. Zhai and B. Liu, "Structured Data Extraction from the Web Based on Partial Tree Arrangement," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 12, pp. 1614-1628, Dec. 2006.

[4 ] 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.

[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 ] P. Bonizzoni and G.D. Vedova, "The Complexity of Multiple Sequence Arrangement with SP-Score that Is a Metric," Theoretical Computer Science, vol. 259, nos. 1/2, pp. 63-79, 2001.

[7 ] D. Buttler, L. Liu, and C. Pu, "A Fully Automated Object Extraction System for the World Wide Web," Proc. 21st Int'l Conf. Distributed Computing Systems, pp. 361-370, 2001.

[8 ] K.C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, "Structured Databases on the Web: Observations and Implications," SIGMOD Record, vol. 33, no. 3, pp. 61 -70, 2004.

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

[10 ] Muslea, S. Minton, and C. Knoblock, "Hierarchical Wrapper Induction for Semistructured Information Sources," Autonomous

[11 ] H. Snoussi, L. Magnin, and J.-Y. Nie, "Heterogeneous Web Data

[12 ] R. Baeza-Yates, "Algorithms for String atching: A Survey," ACM SIGIR Forum, vol. 23, nos. 3/4, pp. 34 -58, 1989.

[13 ] J. Wang and F. Lochovsky, "Data -Rich Section Extraction from HTML Pages," Proc. Third Int'l Conf. Web Information System Eng., 2002.

[14 ] 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.

[15 ] W. Cohen and L. Jensen, "A Structured Wrapper Induction System for Extracting Information from Semi-Structured Documents," Proc. IJCAI Workshop Adaptive Text Extraction and Mining, 2001.

[16 ] A.V. Goldberg and R.E. Tarjan, "A New Approach to The Maximum Flow Problem," Proc. 18th Ann. ACM Symp. Theory of Computing, pp. 136-146, 1986.

[17 ] D. Gusfield, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge Univ. Press, 1997.

[18 ] C. Tao and D.W. Embley, "Automatic Hidden-Web Table Interpretation by Sibling Page Comparison," Proc. 26th Int'l Conf. Conceptual Modeling, pp. 566-581, 2007.

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

[20 ] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith, "Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages," Data and Knowledge Eng., vol. 31, no. 3, pp. 227-251, 1999.