IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 8, Issue 6 (Jan. - Feb. 2013), PP 93-96
www.iosrjournals.org

Structured Data Extraction from the Deep Web

Vimala.S[1], Meenakshi.R[2]
M.E.(Software Engineering)[1], Associate Professor/IT[2]
Saveetha Engineering College, Chennai, India

Abstract: *Online databases called the web databases comprise the deep web. Pages in deep web are dynamically generated on receiving a user query which is submitted through the query interface. Extracting data from these deep web pages automatically are very important in many applications which deal with multiple databases. But extracting structured data from these pages is a challenging problem due to the underlying intricate structures of such pages. Until now, a large number of techniques have been proposed to address this problem, but all of them have several limitations. The proposed work automatically extracts structured data from the deep web by first identifying the data regions and segmenting it into Query Result Records (QRRs) in the web page. It then aligns the QRRs into a table such that data values of the corresponding attributes are put into the same column. The proposed system makes use of both the tag and value data for the alignment to be accurate. It also handles non contiguous QRRs due to presence of auxiliary information and processes nested structures that may exist in the QRRs.*

Keywords– *Data extraction, Deep web, Nested structure, Structured data.*

I. Introduction

The amount of information that is currently available on the internet grows rapidly, making the Web as the largest "knowledge base". Pages in the surface web can be accessed by an unique URL. Whereas traditional search engines cannot see or retrieve the contents of the deep web as those pages do not exist until they are created dynamically as a result of specific search. This deep web is also termed as "hidden" or "invisible" web. The impression has naturally arisen from that data can only be accessed through query interfaces. This "query-only" access mode distinguishes databases on the deep web from the rest of link-based contents (surface web). Extracting data records automatically from these pages enables one to integrate data from multiple websites and web pages to provide value-added services like comparative shopping and meta querying. On giving the user query, web database returns the relevant data encoded in HTML pages. This data is either structured or unstructured. Structured databases provide data objects as structured "relational" records with attribute value pairs. Unstructured databases provide data objects as unstructured media like texts, images, audio and video. In general, a query result page contains not only the actual data, but also other information, such as navigational panels, advertisements, comments etc. This leads to the unstructured content. Only when the extracted data is organized in a structured manner, such as tables, they can be compared and aggregated. Hence, accurate data extraction is vital for applications' correct performance. So these irrelevant information must be eliminated and only the necessary data must be displayed to the user.

II. Related Works

In the recent years, the web has been rapidly deepened by the massive network databases. It is believed that more significant information is hidden in the deep web. Using a searchable database compilation overlap analysis between pairs of search engines, a July 2000 white paper [1] estimated at least 43,761 – 96,702 deep websites and 550 million hidden pages in the deep web which is 500 times larger than the surface web. The total quality content is 1000 to 2000 times greater than that of surface web.

Earlier works on data extraction focused on wrapper induction methods, which requires human assistance to build a wrapper. More recently data extraction methods have been proposed to automatically extract the data records from the query result page. In wrapper induction, a set of data extraction rules are learnt from a set of manually labelled pages [2,3] and it uses them to extract user data. While wrapper induction has an advantage that no extraneous data are retrieved because the user labels the item of interest, it is a time-consuming and labor intensive job. Thus a wrapper defined for a similar set of pages performs poorly when the format of the pages change, which may happen frequently on web. To overcome the problem of wrapper induction some unsupervised learning methods, such as RoadRunner [4] and IEPAD [5] have been proposed to automatically extract the query result records from the web pages. But these methods rely only on HTML tags which is not accurate as it convey little semantic information. Some embedded tags may confuse the wrapper generator which is unreliable.

To overcome these disadvantages a method called ViPER [6] makes use of both tag information and visual data value similarity features. It first identifies and ranks repetitive patterns with respect to user's visual perception of the web page. And then matching patterns are aligned with global multiple sequence alignment techniques. But the main drawback of this method is that it performs poorly with web pages with nested structures. Similar to ViPER a method called ViNT [7] uses both visual similarity and tag structure to extract data records. It first uses visual similarity to find the data value similarity and then combines the tag structure to generate wrappers. But this method is not good because it requires at least 4 QRRs in query result page. If the data records are distributed in many regions in one page then only the large QRR is considered. And when the format of the pages change it is not able to perform well.

To handle nested structures a method called NET [8] has been proposed. This method first builds a tag tree based on visual information. It then performs post-order traversal of the tree and matches the sub trees using visual cues and tree edit distance. The advantage of this method is that it accurately aligns and extracts data records of both flat and nested structures. But the limitation is that it processes the nested structures before the data records are aligned which incorrectly identifies a flat structure as a nested one. A similar technique which uses visual information and tree matching for segmenting the data records, DEPTA [9] has been proposed. It involves a partial alignment technique that aligns the data values onto the corresponding attributes. This method is proved highly effective in large number of web pages from diverse domains. Another similar technique called CTVS [10] has been proposed which outperforms all the previous technique. This method first constructs a tag tree for the HTML page. It then identifies the data regions and segments the data records in it. Then based on the similarity of the records they are merged. Two types of alignment are done here to align the data values onto the correct attributes. At last nested structure processing is done to eliminate the nested structures in the query result.

III. Problem Statement

Data extraction is necessary to extract information from the web. Several semi-automatic and automatic approaches have been reported in the literature for mining data records from the web. Some machine learning approaches which is semi-automatic requires human labelling of the specific regions to mark them interesting. It is labor intensive and time consuming job. Other automatic methods use a set of heuristics and domain ontology to perform the task. Apart from low accuracy, existing systems assume that relevant information is present contiguously in the HTML code and some systems do not handle nested structures. It makes use of only tag information to extract data records which makes them vulnerable to optional attributes problem which makes the tag structure irregular. So to overcome all these problems a system is needed which automatically retrieves the query result from a non contiguous HTML page. It must be able to handle nested structures if present and display the result in structured format to the user.

IV. Proposed System

The proposed system employs 4 steps to extract the QRRs from the query result page. When a user inputs the query to be searched to the query interface it displays the web page. Assuming that there are at least 2 QRRs in a query result page, the system first constructs a tree for the HTML page. Then the data regions are identified using a top-down approach. The data region are segmented into data records based on the similarity between the attributes. Then an alignment is done so that the data values from the same attributes are put into the same column. Finally a nested structure processing is done to identify any nested structures. The architecture of this system is given in Fig. 1.

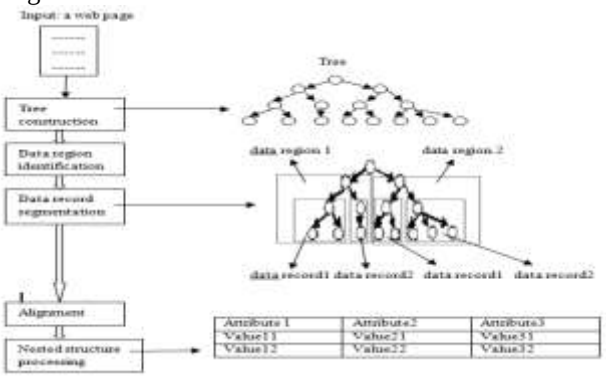

Fig. 1. The general architecture

4.1. Tree construction

In a Web browser, each HTML element (consisting the start tag, optional attributes, text, end tag) is rendered as a rectangle. Each element is a node in the tree. Before creating the tree the formatting tags like <B>

</B>, <I> </I> etc are removed. First the four boundaries of the rectangle for each HTML element is found with the help of embedded HTML rendering engine of the Web browser. Then we have to check whether one rectangle is enclosed within other rectangle or not. Using that information we can create the tree for the web page. The inner rectangle forms the the child of the enclosing rectangle.

4.2. Data region identification

A data region is an area in a page that contains similar data records. Instead of mining the data records directly, first the data regions are found and then the data records within them are found. First we assume that some child subtrees of the same parent form similar data records which combine to form a data region. Many existing systems assume that these data records are present contiguously in a web page. But some unwanted information are also present and we have to eliminate them. Data region identification algorithm uses a top-down approach. Starting from the root node of the tree this algorithm is applied recursively to all its children. The two steps in this algorithm are: 1) Calculate the similarity of each pair of nodes. If the value is greater than threshold 0.6, as given in many of the existing systems, then the two nodes are similar. This is applied to all its children and all the similar nodes form a data region. 2) Segment the data regions into data records by the record segmentation algorithm.

4.2.1. Data record identification

Since a data region contains several data records, similar nodes are assigned a number. For ex. let a data region be represented as 1212121. Here there are 2 repeats, 12 and 21. Auxiliary information may be present in any part of the data region. The repeat that stops at the auxiliary information is the correct repeat and it is considered as one data record. Similarly all the data records within a data region are found. It is also found that the distance between data records is larger than the distance within a data record.

4.3. Alignment of data values

The data records found from the above step goes to the alignment process. Among several sub trees (data records), the sub tree that contains maximum number of fields is chosen as the head tree, denoted by T_h. Other trees are denoted by $T_i (i \neq h)$. A match is found for each node in T_i with that of nodes in T_h. If a match is found then a link is created between the two nodes indicating a match in the head tree. If no match is found then the node is just inserted into T_h and the tree is expanded. This is illustrated in the Fig. 2.

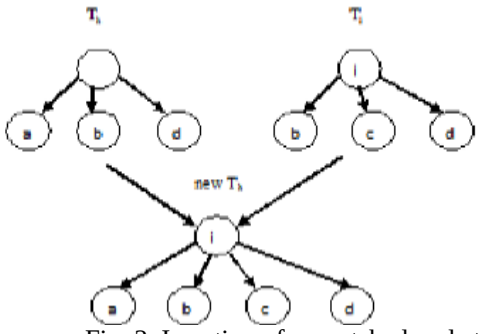

Fig. 2. Insertion of unmatched node to the head tree

Here the nodes b and d of T_i are matched with nodes in T_h, so a link is established between them. But the node c in T_i is not matched with any of the nodes in T_h since it may contain some extra data items. Hence it is just inserted into T_h. Thus the tree now contains 4 data items.

4.4. Nested structure processing

Nested structures are due to an attribute having multiple values. So some of the data values may not be aligned to other values. The nested structure processing algorithm first identifies nested columns in the QRR. After all the nested columns are identified, a new row is created by copying the repetitive data parts and the remaining parts as well. The nested structure processing algorithm is given in Fig. 3.

```
Function nest_proc (QRRs, T, align)
1:  N ← ∅
2:  for each QRR in T with root t
3:          nest_column_find (t, T, align, N)
4:  for each nested column n in N do
5:          create a new row for each repetitive part
Function nest_column_find (t, T, align, N)
6:  if ( t contains more than one data value) then
7:  for each subtree t_i of t do
8:          nest_column_find (t_i, t, align, N)
9:  if (data values are repeated in t) then
10:         n_p is a nested column
11:  if n_p does not belong to N then
12:  add_column (n_p, N)
Function add_column (n_p, N)
13:  for each node n_i in N do
14:         if (n_p ∩ n_i ≠ ∅) then
15:           N← N ∪ n_i
16:         break
```

Fig. 3. Nested structure processing algorithm

Given a tree T and QRRs, the function nest_proc works as follows. Let the nested column set be denoted by N. In Fig. the nested column set is initialized to NULL (line 1). For each QRR with root t in T the nest_column_find finds any nested column if present(line 2 and 3). A new row is created for each nested column (line 4 and 5). In the function nest_column_find for each subtree t_i of t the function is repeated (line 6-8). If data values are repeated then it is considered as a nested column n_p and inserted into the nested column set N (line 9-12). The function add_column adds the nested column n_p into N if it is not already present in the nested column set (line 13-16). This algorithm makes use of both tag and value information which identifies the nested structures correctly. Whereas many existing systems make use of only tag information which may incorrectly identify a flat structure as a nested one.

V. Conclusion

Thus we proposed a system for automatic extraction of structured query results from a deep web page. It is able to eliminate all the auxiliary information in the web page. And it is also able to handle non contiguous web pages which is not done in the existing techniques. The proposed system includes 4 steps: First it constructs a tree from the HTML page. Second the data regions that contains the query results are identified. The data records inside the regions are segmented. Third the data values are aligned to their corresponding attributes. Finally all the nested structures are found and processed. This system extracts the result using both the tag and value information which the existing systems does not do. This gives more accurate results and the alignment is made easier. This system is very useful in many web applications that needs data from multiple websites in less time.

References

[1] BrightPlanet.com. The deep web: Surfacing hidden value. Accessible at http://brightplanet.com, July 2000.

[2] I. Muslea, S. Minton, and C. Knoblock, "Hierarchical Wrapper Induction for Semistructured Information Sources", *Autonomous Agents and Multi-Agent Systems*, vol. 4, nos. ½, pp. 93-114, 2001.

[3] 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.

[4] 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.

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

[6] 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.

[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 Extracting Web Data from Flat and Nested Data Records",. *Proc. 6th Int'l Conf., Web Information Systems Engg*, pp. 487-495, 2005.

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

[10] Weifeng Su, Jiying Wang, Frederick H. Lochovsky and Yi Liu, "Combining Tag and Value Similarity for Data Extraction and Alignment", *IEEE Trans., Knowledge and Data Engg.*, vol. 24, no. 7, pp. 1186-1200, Jul. 2012.