ISSN:0975-9646

Search Results From the Web Databases Using Ontology-Assisted Data Extraction

J. Siva Jyothi , Ch. Satyananada Reddy

Department of CS&SE, Andhra University
Visakhapatnam, Andhra Pradesh, India.

***Abstract*— With the help of HTML form-based search interfaces, a large number of databases have become web accessible. For the sake of human browsing, the data units from the underlying database are decoded into the result pages very dynamically. The encoded data units need to be machine processable. It is very important for many applications such as deep web data collection and Internet applications, and are extracted and meaningful labels are assigned. It is accessible data extraction method, ODE (Ontology-assisted Data Extraction), which automatically extracts the query result records from the HTML pages. ODE first constructs ontology for a domain according to information matching between the query interfaces and query result pages from different web sites within the same domain. Then, the constructed domain ontology is used during data extraction to identify the query result section in a query result page and to align and label the data values in the extracted records. The ontology assisted data extraction method is fully automatic and overcomes many of the deficiencies of current automatic data extraction methods. The annotation wrapper which is automatically constructed for that particular site can be used to annotate new result pages from the same web database. By the test results it is found that this approach is good and effective.**

Keywords— **Data alignment, data annotation, web database, wrapper generation**

I. INTRODUCTION

Generally web is a data base i.e., the returned result pages of any search engine come from the structured database. Such type of search engines is often referred as Web databases (WDB). A response page generated from a WDB has many search result records (SRRs). For every SRR, it contains numerous data units which define one feature of a real time object.

The main challenge now-a-days is to generate metadata by huge combinations. For this perspective, a main technique for this type of problem is known as "Deep Annotation", which uses three characteristics of data—the data itself, its arrangement, and its framework—to derive mappings. For the online databases, they respond to the user query with the output data embedded in HTML file. To get the records from the HTML files automatically, a technique implemented i.e. Data extraction. A new methodology which extracts data from HTML pages is "ODE (Ontology-assisted Data Extraction)".

In general, ODE first makes ontology for a domain allowing to info similar among the query interfaces and query result pages from different web sites within the same area. Then, the constructed domain ontology is used during data extraction to identify the query result section in a query result page and to align and label the data values in the extracted records.

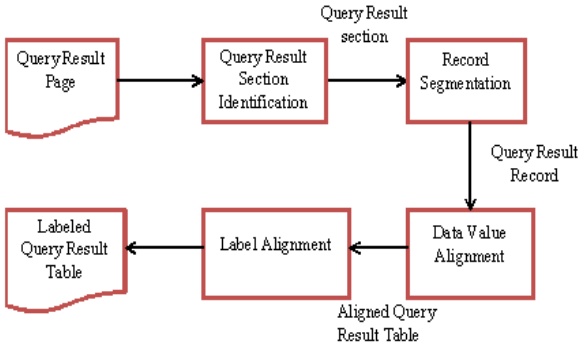

Figure 1: Data Extraction

For example, once an electrical product comparison shopping system collects numerous consequence records from different electronics websites. It needs to decide whether any two SRRs denote to the similar electrical product. To obtain this, the item numbers have to be compared. Here the prices have to be listed in order by each site. Then, the classifications have to know the semantic of each information element. For later analysis, these collected SRR's has been stored and which contains semantic labels for data units is not only important for the record linkage task. In previous applications, basically it requires severe human determinations to annotate data units manually, which hardly bound their scalability. In the proposed approach it is studied how to automatically allocate tags to the data units within the SRRs reimbursed from WDBs. The search result records for a result page are signified in table arrangement with each row representing an SRR.

The Proposed System consists of three phases i.e. alignment phase, annotation phase and annotation wrapper generation phase. Phase 1 is the alignment phase where it has to classify all data units in the SRRs and then form them into dissimilar collections with each cluster related to a diverse conception. Phase 2 is the annotation phase where it is present table annotator which is used to create a tag for the items within their collection, and a possibility model is

accepted to define the most suitable tag for each set. For the recognized notion, it is create an annotation instruction that defines how to abstract the data units of this concept in the outcome page and what the suitable semantic label.

In our paper, a normal data element is a part of text that systematically represents one concept of a real time object. It corresponds to the value of a record under an attribute. This is entirely a diverse from a text node which denotes to an arrangement of text enclosed by a couple of HTML tags. Combined data units of the same group can help to recognize the mutual forms and features among these data units. These similar characteristics are the root for our annotators. The rules for all aligned groups, collectively, form the annotation wrapper for the corresponding WDB, which can be used to directly annotate the data retrieved from the same WDB in response to new queries without the need to perform the alignment and annotation phases again. As such, annotation wrappers can perform annotation quickly, which is essential for online applications.

The remaining paper will be as follows: Section II and Section III contains the Background knowledge to do this work and Related Work. Section IV includes the proposed method for the already existed problem. Section V describes how this paper has been implemented and been used. Section VI defines the conclusion of the paper and the future enhancement what are going to be done.

II. BACKGROUND

The main aim of extraction of data is to reduce the unwanted data from a query output page, get the query result records (referred to in this paper as *QRRs*) from the page and arrange the obtained values in the extracted records into a table. Then the data standards for the identical attribute in each record are kept into the similar column in the table.

1. *Query result sectional identification* decides what section in a dynamically generated query result page contains the data that need to be extracted.
2. *Record segmentation* segments the query result section into records and extracts them.
3. *Data value alignment* aligns the data values2 from multiple records that belong to the same attribute so that they can be arranged into a table.
4. *Label assignment* assigns a suitable, meaningful label (i.e., an attribute name) to each column in an aligned table.

Data extractions based determine is also commenced to evaluate the importance of each leaf chunk in the tree, which in turn helps us to get rid of noises in a deep Web page. In this determine, eliminate the excess clatter and duplicate chunk using three parameters such as hyperlink percentage, Noise score and cosine similarity. Lastly, gets the main chunk mining process using three parameters such as Title word Relevancy, Keyword frequency based chunk selection.

In this project, to arrange all the data units and explain the ones within the similar semantic cluster holistically. Here alignment of the resultant data is a vital pace in obtaining correct explanation. Many of the presented routine data alignment methods are based on not a single feature, but also on more. Frequently used characteristic is HTML tag paths. In this, the sub trees related to two different data units in dissimilar SRRs but which have the same concept i.e., having the same tag structure. On the other hand, it can't say that this assumption is not always accurate as the tag tree is very responsive to even slight dissimilarities, which may be occurred by the need to highlight certain data units or erroneous coding.

Generally, in our project, the alignment of the data is different from the early works in the respective aspects. The initial step is that our process switches the associations among text nodes and data units, while previous techniques regard as only some of the types. After this, it was used a different selection of characteristics together, together with the ones used in previous techniques, even as the previous technique has some features only. The features are used automatically received from the output page and no need of any domain specific idea. Finally, to introduce a new clustering-based shifting algorithm to perform alignment.

III. RELATED WORK

In present days, the quality and the volume of web information has involved much research attention. As the returned data for a query is embedded in HTML pages, much research has focused on extracting the data from these query search result records. Simultaneously, many researchers have studied the problem of mining information from HTML data files. Previous work focused on the wrapper induction, during which human assistance is required to build the wrapper.

Embley et al [10] utilize ontology and other heuristics to automatically extract data in multiple records and label them. But ontology for different domain needs to be constructed manually. Arasu et al [1] describe about extracting structured data from the web page. In which structured template is used to extract the information from the web page. To extract information from the unstructured page structured template pages are used. The human input is absent here so that the occurrence of error is limited and time consuming. But it does not suitable for large database also it does not say about crawling, indexing and providing support to querying structure pages in web.

This paper is about the relationships between text nodes and data units. Specifically, this paper identifies four relationship types and provides analysis of each type, while only two of the four types (i.e., one-to-one and one-to-many) are very briefly mentioned in [13]. A new step is added to handle the many-to-one relationship between text nodes and data units. The experiment section (Section V) is significantly different from the previous description. The data set used for experiments has been extended by one domain WDBs. Moreover, the experiments on alignment and annotation have been rebuilt based on the new data set and the alignment algorithm.

IV. PROPOSED METHOD

In this proposed method, many-to-one relationship between the text nodes and data units is additionally taken. In this case, multiple text nodes together form a data unit. For this purpose, it needs to extract and annotate to identify and remove decorative tags inside search results so that the completeness of each split data unit can be restored. The first step of our alignment algorithm handles this case specifically.

The ontology for a result record is first constructed from query result pages and query interfaces of the web sites, and then used to extract data records from a query result page in the domain. In this section, to present the observations about query result pages and query interfaces of web databases, on which the ontology construction approach is based. The ontology provides an overview of the Query Result Records(QRRs) generated from a web database, the influence of optional attributes on the data extraction is decreased, especially for the data value alignment.

The search results are obtained in web pages have been given as input to the system. Then these search results are executed in the first phase known as dividing the data into groups for arrangement and then in second phase, annotation will takes place. At last in third phase, it mainly concentrates on annotation wrappers those supplies concluding annotated web pages by applying two kinds of annotators: Table Annotator (TA) and Query Based Annotator (QA).

Table Annotator:

All the search engines represent their data in the tabular format which is easy to access. This means the obtained search results will be represented in the tabular arrangement. The data in tabular format can help users to understand it by a glance. The main point of this Table Annotator is to identify the field headers in the table. After identifying the headers, the data items will be processed. When the processing is completed, the vertical overlie in a column is recognized and then for labelling, header text is used.

Query based Annotator:

This Query based annotator considers the basic search results of a processed query are related to the same query only. To annotate the data items, Name of the search column label has been used. In the database, it doesn't contain all the attributes. To avoid this situation, query based annotator is most helpful in this context.

The proposed method framework consists of two steps:

4.1 Alignment Algorithm

4.2 Clustering Algorithm

4.1. Alignment Algorithm:

Although the SRRs have different clusters of attributes, our data alignment algorithm is supported on the assumption that attributes materialize in the similar order over all SRRs on the identical result page. This is a fact that the SRRs from the similar WDB are generally created by the similar template curriculum. Then, it can theoretically think about the SRRs on an output page in a table format. In this, each row characterizes one SRR and each cell seizes a data unit. In our paper, the column in the tables is referred to as an alignment group which contains at most one data unit from every SRR. If the generalized alignment group contains all the data items of single concept and not having any data item from other concepts, then it is referred to be as well-aligned group. The main concept of this paper is to move generalized data items in the every table so that each and every group is well aligned, while the categorize of the data items within each SRR is stored. The data alignment method consists of the following four steps.

Step 1: Merge text nodes: In this step, first it identifies and eliminates attractive tags from every SRR to permit the text nodes equivalent to the identical attribute to be combined into a single text node.

Step 2: Align text nodes: After merging the text nodes, this aligns the resultant text nodes into groups so that finally each collection has the text nodes with the similar concept or the identical set of concepts.

Step 3: Split (composite) text nodes: A collection whose "values" need to be split into different pieces is called a composite group. After aligning the text nodes, then it have to split the "values" in composite text nodes into separate data units. This step is done perfectly when the text nodes belong to same group only.

Step 4: Align data units: Final step is to divide every composite group into multiple aligned collections which is having data units of the similar concept

4.2 Clustering Algorithm:

CLUSTERING (G) describes that:

1. Read all fully available records from the annotation stage.
2. For each record evaluate all the "child node", and if child nodes contain full data then those records will be taken high distance records.
3. Non–available "child nodes" will be pushed in to the last part of Result Records (RR).

After the completion of grouping using the algorithm, to identify whether that group is needed to split once again to get the required data units. For this, to identify the groups which are not split-able. For this, every group have to satisfy any one of the following conditions given below:

1. Group having the text nodes must be a hyperlink;
2. Group contains nodes; where text of those nodes is similar.
3. If the nodes in the group contains same non-string data type.
4. If that group is not a merged one i.e., single group.

Due to the missing values in the composite text node, the group is not always aligned after splitting into

parts. For this, suggest a solution that to repeat the same alignment algorithm i.e., because in starting the alignment is depended on every data unit's usual place and then have to apply the clustering-based shifting method. The simple difference among this is that, because the data units which have to be aligned are splitted from the identical composite text node, which shares the similar appearance style and tag path. For this, the two features are not helpful for calculating comparison for aligning data units.

ALIGN (RRs):

```
1.  b←1;
2.  While true
3.  for a←1 to number of RRs
4.  GRP_b←RR[a][b];
5.  if GRP_b is empty
6.  exit;
7.  V←CLUSTERING(GRP);
8.  if  |V| >1
9.  S←Ø;
10. for P← 1 to number of RRs
11. for Q←b+1 to RR[a] length
12. S< RR[p][q];
13. V[c]=min(sim(V[c],S));
14. for c←1 to |v| and c≠c
15. foreach RR[p][b] in V[C]
16. insert NIL at position b in RR[p];
17. b←b+1;
```

Algorithm for aligning the data units

CLUSTERING (GRP):

```
1.  V←all data units in GRP
2.  while |V| >1
3.  best←0;
4.  L←NIL; R←NIL;
5.  foreach X in V
6.  foreach Y in V
7.  if((X!=Y) and (sim(X,Y)>best))
8.  best←sim(X,Y);
9.  L←X;
10. R←Y;
11. if best > T
12. remove L from V;
13. remove R from V;
14. add L U R to V;
15. else break loop;
16. return V;
```

Algorithm for clustering the aligned data units

V. IMPLEMENTATION AND EVALUATION

Data Sets and Performance Measure

The implementation process of an operational prototype of WISE-iExtractor[12] using Java. WISE-iExtractor proceeds raw HTML pages as input containing search interfaces (search forms containing scripts such as JavaScript or VbScript are not considered) in the same domain, and outputs the schemas of these interfaces in XML format for use by other applications (schema matching).The total extraction process is fully automated.

To evaluate the interface extraction technique proposed in this paper, it is selected various numbers of search interfaces from five application domains: books, electronics, games, music and vehicles.

Experimental Results

The proposed system performance is evaluated on the basis of two factors: precision and Recall. The precision and recall is calculated for performance of alignment. The precision for performance of alignment is as follows.

$$\text{Precision} = \frac{\text{Correctly Aligned Data units}}{\text{Aligned Data Unit}} \times 100$$

$$\text{Recall} = \frac{\text{Data units that are Correctly Aligned}}{\text{Manually Aligned Data Unit}} \times 100$$

The optimal feature weights obtained through the method[11] over data set is {0.7, 0.9, 1.0,0.5, 0.6}and 0.59 for clustering threshold T. The average alignment precision and recall are converged at about 97.8%. The learning result shows the data type and the presentation style is the most important features in our alignment method. Then, it applies our annotation method on first Dataset to determine the success rate of each annotator.

Figure.2 shows the performance of our data alignment algorithm for all 90 pages in Second Dataset. The precision and recall for every domain are above 97%, and the average precision and recall across all domains are above 98%. The performance is consistent with that obtained over the training set. The errors usually happen in the following cases. First, some composite text nodes failed to be split into correct data units when no explicit separators can be recognized.

The experiments data from various domains with respect to two annotators only.The annotators used include table annotator and query–based annotator. Both the annotators are supported by the prototype application and it is extensible so as to support more annotators in future. The performance of data alignment and annotation are presented in table.

Domain	Alignment	
	Precision	**Recall**
Books	98	97
Electronics	98.4	97.3
Games	99	98.2
Music	98.4	98
Vehicles	97.5	97.1
Average	**97.96**	**97.79**

Table1. Experimental Results

As presented in Table1, it is evident that more than 97% precision and recall were recorded for both the performances such as data alignment and annotations. The table also shows the performance of annotation with wrapper. The results are presented in the following graph. As shown figure 2, it is evident that the prototype application is capable of producing annotations automatically given search results of Google. The performance of the application is encouraging and the application can be used in the real world applications.

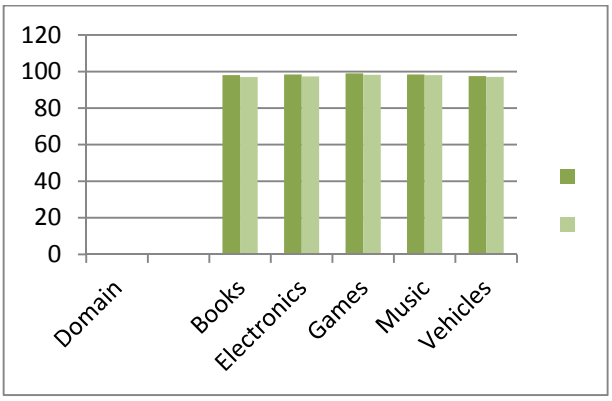

Figure 2: Performance of data alignment for domains

VI. CONCLUSION

Generally, in ODE, first the ontology has been designed for a domain by corresponding the query boundaries and the query result pages by taking from the diverse web sites. After that only, the ontology which has been designed is used to do the data extraction. In general, ODE uses a greatest entropy replica for data value arrangement and label task. Context, tag structure and visual information are used as features for the maximum entropy model.

Here the data annotation problem is studied and proposed a multi annotator approach to automatically constructing an annotation wrapper for annotating the search result records retrieved from any given web database. The application visualizes results which are nothing but the annotated documents. HTML tags are used to process the pages while annotating them. The annotated results are further useful in real world applications. The empirical results revealed that our application is effective.

In this paper, to study how the data will be aligned automatically, it is critical to achieve accurate alignment. The method is used and it is a clustering based shifting method which utilizes automatically accessible features. By this method, it is very much capable of handling different types of relationships between HTML text nodes and data units.

REFERENCES

[1] A. Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages," Proc. SIGMOD Int'l Conf. Management of Data, 2003.

[2] L. Arlotta, V. Crescenzi, G. Mecca, and P. Merialdo, "Automatic Annotation of Data Extracted from Large Web Sites," Proc. Sixth Int'l Workshop the Web and Databases (WebDB), 2003.

[3] W. Su, J. Wang, and F.H. Lochovsky, "ODE: Ontology-Assisted Data Extraction," ACM Trans. Database Systems, vol. 34, no. 2, article 12, June 2009.

[4] J. Lee, "Analyses of Multiple Evidence Combination," Proc. 20th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, 1997.

[5] L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled WrapperConstruction System for Web Information Sources," Proc. IEEE16th Int'l Conf. Data Eng.,2001.

[6] W. Liu, X. Meng, and W. Meng, "ViDE: A Vision-Based Approach for Deep Web Data Extraction," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 3, pp. 447-460, Mar. 2010.

[7] Y. Zhai and B. Liu, "Web Data Extraction Based on Partial Tree Alignment," Proc. 14th Int'l Conf. World Wide Web (WWW '05),2005.

[8] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu,"Fully Automatic Wrapper Generation for Search Engines," Proc. Int'l Conf. World Wide Web, 2005.

[9] H. Zhao, W. Meng, and C. Yu, "Mining Templates form Search Result Records of Search Engines," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2007.

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

[11] Yiyao Lu ,Hai he, " Annotating Search Results from Web Data bases" IEEE Trans. Knowledge and Data Eng., vol. 25, no. 3, Mar. 2013.

[12] H. He, W. Meng, C. Yu, and Z. Wu,"Constructing Interface Schemas for Search Interfaces of Web Databases," Proc. Web Information Systems Eng. (WISE) Conf., 2005.

[13] Y. Lu, H. He, H. Zhao, W. Meng, and C. Yu,"Annotating Structured Data of the Deep Web," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), 2007.