Enhanced Search Results with Semantic Annotation Approach

*S. Anuradha, G. Bhavana, Ch. D. V. Sudheer, G. Suryanarayana, G. Govind

Department of CSE, VITS College of Engineering, Visakhapatnam, INDIA, *anu.surabhi1980@gmail.com

Abstract- An increasing number of databases have become web accessible through HTML form based search interfaces. The data units returned from the underlying database are usually encoded into the result pages dynamically for human browsing. For the encoded data units to be machine processable, which is essential for many applications such as deep web data collection and Internet comparison shopping, they need to be extracted out and assigned meaningful labels. In this paper, we present an automatic annotation approach that first aligns the data units on a result page into different groups such that the data in the same group have the same semantic. Then, for each group we annotate it from different aspects and aggregate the different annotations to predict a final annotation label for it. An annotation wrapper for the search site is automatically constructed and can be used to annotate new result pages from the same web database. Our experiments indicate that the proposed approach is good and effective.

Keywords- data alignment, data annotation, data unit, search result record, search pattern, semantic, text node, wrapper generation

1 INTRODUCTION

Generally web is database based, 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 typical result page returned from a WDB has multiple search result records (SRRs). Each SRR contains multiple data units each of which describes one aspect of a real world entity. Fig. 1 shows three SRRs on a result page from a mobile store WDB. Each SRR represents a mobile with several data units, e.g., the first mobile record has data units "Sony Xperia M - Flipkart www.flipkart.com ,Mobiles & Accessories, Mobiles Rating: 4.4 - 1,260 votes, Rs. 10,750.00 Buy Sony Xperia M for Rs.10700 Online, Also get Sony Xperia M Specs & Features. Only Genuine Products. 30 Day Replacement Guarantee. Free Shipping".

Collecting data of interest from multiple WDBs is increasing nowadays [1]. For example, once a mobile product comparison shopping system collects multiple result records from different mobile websites, it needs to determine whether any two SRRs refer to the same mobile. The item number can be compared to achieve this. The system also needs to list the prices offered by each site. Thus, the system needs to know the semantic of each data unit. In Fig. 1, no semantic labels for the values of title, rating, price, etc., are given. Having semantic labels for data units is not only important for the above record linkage task, but also for storing collected SRRs into a database table for later analysis [1]. Early applications require tremendous human efforts to annotate data units manually, which severely limit their scalability. In this paper, we consider how to automatically assign labels to the data units within the SRRs returned from WDBs. The SRRs of a particular result page are returned from a WDB, our automatic annotation solution consists of three phases as illustrated in Fig. 2.

Fig. 1 Original HTML Page

Let d_{ji} denote the data unit belonging to the i^{th} SRR of concept j.The SRRs of a result page are represented in table format, as in Fig. 2a, with each row representing an SRR. Phase1 is the alignment phase. In this phase, we first identify all data units in the SRRs and then organize them into different groups with each group corresponding to a different concept (e.g., all titles are grouped together). Fig. 2b shows the result of this phase with each column containing data units of the same concept across all SRRs [2]. Grouping data units of the same semantic can help to identify the common patterns and features among these data units. These common features are the basis of our

annotators. In Phase2, the annotation phase, we introduce table annotator which is used to produce a label for the units within their group holistically, and a probability model is adopted to determine the most appropriate label for each group. Fig. 2c shows that at the end of this phase, a semantic label L_j is assigned to each column. In Phase3, the annotation wrapper generation phase, as in Fig. 2d, for each identified concept, we generate an annotation rule R_j that describes how to extract the data units of this concept in the result page and what the appropriate semantic label should be[3], [4]. 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.

2 FUNDAMENTALS

2.1 Web Database

A web database is an organized listing of web pages. It's like the card catalog that you might find in the library. The database holds a "surrogate" (or selected pieces like the title, the headings, etc.) for each web page. The creation of these surrogates is called "indexing", and each web database does it in a different way. Web databases hold surrogates for anywhere from 1 million to several billion web pages. The program also has a search interface, which is the box you type words into (like in Alta Vista or Google) or the lists of directories you pick from (like in Yahoo). Thus, each web database has a different indexing method and a different search interface.

2.2 *Search Pattern*

A Search Pattern is a system which is used in a search to facilitate the most efficient, effective, and successful search possible. Several companies design software which can be used to devise search patterns, and it is also possible to map one out by hand.

2.3 Annotation

Annotations are comments, explanations, notes or other types of external remarks that can be attached to a web document or to a selected part of document. As they are external, it is possible to annotate any web document independently, without needing to edit the document itself. From the technical point of view, annotations are usually seen as metadata, as they give additional information about an existing piece of data.

2.4 Wrapper

Wrapper is a cross-browser compliant HTML/CSS rendering engine written in Action Script that sits on top of your standards compliant HTML page. Wrapper eliminates cross-browser issues and makes integrating Action Script and HTML/CSS projects possible without needing to compile every change.

2.5 Alignment

The align attribute defines the vertical or horizontal alignment of various multimedia elements. This attribute is valid in HTML 4, but not in HTML 5. We should use CSS to align your elements instead.

2.6 Data Unit and Text Node

Each SRR extracted has a tag structure that determines how the contents of the SRRs are displayed on a web browser. Each node in such a tag structure is either a tag node or a text node. A tag node corresponds to an HTML tag surrounded by "<" and ">" in HTML source, while a text node is the text outside the "<" and ">." Text nodes are the visible elements on the webpage and data units are located in the text nodes. However, as we can see from Fig. 1, text nodes are not always identical to data units. Since our annotation is at the data unit level, we need to identify data units from text nodes.

2.7 Data Unit and Text Node Features

We identify and use five common features shared by the data units belonging to the same concept across all SRRs, and all of them can be automatically obtained. It is not difficult to see that all these features are applicable to text nodes, including composite text nodes involving the same set of concepts, and template text nodes.

2.7.1. Data Content (DC)

The data units or text nodes with the same concept often share certain keywords. This is true for two reasons. First, the data units corresponding to the search field where the user enters a search condition usually contain the search keywords. For example, in Fig. 1, the sample result page is returned for the search on the title field with keyword "sony xperia." We can see that all the titles have this keyword. Second, web designers sometimes put some leading label in front of certain data unit within the same text node to make it easier for users to understand the data. Text nodes that contain data units of the same concept usually have the same leading label. For example, in Fig. 1, the price of mobile has leading words "Buy Sony Xperia for" in the same text node.

2.7.2 Presentation Style (PS)

This feature describes how a data unit is displayed on a webpage. It consists of six style features: font face, font size, font color, font weight, text decoration (underline, strike, etc.), and whether it is italic. Data units of the same concept in different SRRs are usually displayed in the same style. For example, in Fig. 1, all the availability information is displayed in the exactly same presentation style.

2.7.3. Data Type (DT)

Each data unit has its own semantic type although it is just a text string in the HTML code. The following basic data types are currently considered in our approach: Date, Time, Currency, Integer, Decimal, Percentage, Symbol, and String. String type is further defined in All-Capitalized-String, First-Letter-Capitalized-String, and Ordinary String. The data type of a composite text node is the concatenation of the data types of all its data units. For example, the data type of the text node "Rating: 4.4 - 1,260 votes Rs. 10,750.00 Buy Sony Xperia M for Rs.10700" in Fig. 1 is <First-Letter-Capitalized-String> <Symbol> <Symbol> <Integer> <Symbol> <Integer> < String> <Integer> <Decimal> <Integer>. Consecutive terms with the same data type are treated as a single term and only one of them will be kept. Each type except Ordinary String has certain pattern(s) so that it can be easily identified. The data units of the same concept or text nodes involving the same set of concepts usually have the same data type.

2.7.4. Tag Path (TP)

A tag path of a text node is a sequence of tags traversing from the root of the SRR to the corresponding node in the tag tree. Each node in the expression contains two parts, one is the tag name, and the other is the direction indicating whether the next node is the next sibling (denoted as "S") or the first child (denoted as "C"). Text node is simply represented as <#TEXT>. An observation is that the tag paths of the text nodes with the same set of concepts have very similar tag paths, though in many cases, not exactly the same.

2.7.5. Adjacency (AD)

For a given data unit d in an SRR, let dp and ds denote the data units immediately before and after d in the SRR, respectively. We refer dp and ds as the preceding and succeeding data units of d, respectively. Consider two data units d1 and d2 from two separate SRRs. It can be observed that if dp 1 and dp 2 belong to the same concept and/or ds 1 and ds 2 belong to the same concept, then it is more likely that d1 and d2 also belong to the same concept. We note that none of the above five features is guaranteed to be true for any particular pair of data units (or text nodes) with the same concept. However, such data units usually share some other features.

3 OVERVIEW OF THE SYSTEM

This system architecture is broadly classified into three major phases: Alignment phase, Annotation phase, and Annotation Wrapper Generation phase.

3.1 Data Record Extraction

For the annotation to be done, the data records or the SRRs have to be extracted from the result pages. That is the irrelevant information like advertisements, links, information about the hosting site are to be discarded from the result page. Manually writing programs to extract the records from the result page is laborious, time consuming and impractical since the search engine change the display of the result page time to time. So, a system called ViNTs (Visual information aNd Tag structure based wrapper generator) described in [5] is employed for extracting the search records from the results page. ViNt system has its own architecture for extracting SRRs. The data extraction through ViNt is bases on the visual content features of the web page and also the HTML tag structure which is nothing but the source file of the page in HTML format.

3.2 Data Alignment

The data units are not aligned whenever the search result records are extracted from the web page. The main purpose of data alignment is to group the data units from different records into the semantically same group. This alignment of data records facilitates easier annotation of data. It is based on the assumption that the data units in different SRRs of the same semantic usually have the fixed layout and presentation. Based on this assumption a record expression (REXP) [6], [7] is constructed for each result record. An REXP is a string comprising of sequence of symbols that represents either the presentation style of the node or the separator/ delimiter. For example, in our current implementation of REXP, \S" denotes a pure text node with bold style, \s" denotes a pure text node without bold style, \L" denotes a link node with bold style, \l" denotes a link without bold style, \^" denotes starting a new line, etc. Separators are nodes that contain only non-letter and non-digit characters appearing in HTML text. The REXP for each SRR can be constructed easily. Example 1: The REXP of the first record in Figure 1 is \l^ss/s/s/s^sS»s^s", where \=" and \»" are the separators appearing in HTML text of the record. Note that as shown the text \Peter J. Denning" and a \=" are en-coded together, so the first \s" in the REXP represents \Peter J. Denning=".\Put in Basket" is not included because buttons, icons and images are currently ignored. The last \s" represents \Out-Of-Stock".

With this Record expression a suffix tree is constructed. Then the most common longest string (MCLS) is selected and its corresponding components are aligned to form groups. The data Alignment also concentrates on the Data Unit Similarity, Data Content Similarity, Presentation Style Similarity, Data Type Similarity, Tag Path Similarity as described in [8]. Also for improving the efficiency of data grouping and aligning, Alignment and cluster-shifting technique is also used. The algorithm concentrates on four steps namely merging text nodes, Aligning text nodes, Splitting Composite Text nodes and Align Data Units. The alignment algorithm is shown in figure 4.

3.3 Data Annotation

The data annotation is based on the concept that the data units corresponding to the same attribute always share some common features. These common features are the basis of our annotators. There are six basic annotators used for annotating the database namely Table annotator, Query- based annotator, Schema annotator, Frequency- based annotator, Prefix/Suffix annotator and Common annotator as described in [9], [10]. Many WDBs use a table to organize the returned SRRs. In the table, each row represents an SRR. The table header, which indicates the meaning of each column, is usually located at the top of the table. Fig. 2 shows an example of SRRs presented in a table format. Usually, the data units of the same concepts are well aligned with its corresponding column header. This special feature of the table layout can be utilized to annotate the SRRs. Then annotation and wrapper generation are applied to obtain the wrapper record.

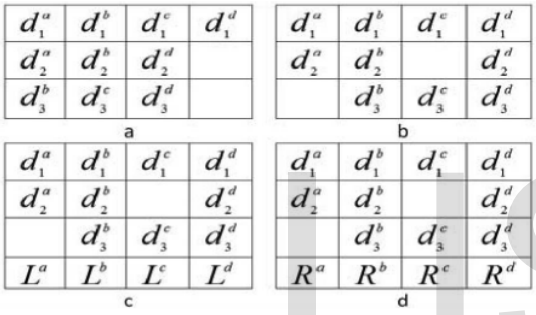

Fig. 2 Illustration of three phase annotation solution

3.4 Automatic Wrapper Generation

Once the data units have been annotated, annotation wrapper is constructed as in [11], [12]. Annotation wrapper consists of certain rules that can be used for new queries without repeating the entire process again.

4 ALGORITHM DEVELOPMENT

4.1 Algorithm for Getting Data Records:

ALIGN_SRR describes that

1. Read Source Document which contains All Records in Xml Node Format.

2. Process each record in Xml nodes in the document one by one.

3. For each "Node" in the "Root", check whether the element contains data or empty node.

4. If element contains "data node", then we are going to consider them as fully qualified records, which can be used in accessing for search process.

5. If element doesn't contain "data node", which might be missing in construction of document which is of no use we are going to eliminate them for further processing.

4.2 Algorithm for Getting SRR's:

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 SRR generation.4.
4. When user performs "search" SRR's will be processed according to fully available data.

ALIGN_SRR

```
1.  j ← 1;
2.  while true
      //create alignment groups
3.  for i ← 1 to number of SRRs
4.  Gi ← SRR[i][j];        //jth element in SRR[i]
5.  if Gi is empty
6.  exit;  //break the loop
7.  V ← CLUSTERING(G);
8.  if |V| > 1
      //collect all data units in group following j
9.  S ← ∅;
10. for x ← 1 to number of SRRs
11.   for y ← j+1 to SRR[i].length
12.     S ← SRR[x][y];
    // find cluster c least similar to following groups
13. V[c] = min k=1 to |V| (sim(V[k], S));
    //shifting
14. for k ← 1 to |V| and k ≠ c
15.     for each SRR[x][j] in V[k]
16.       insert NIL at position j in SRR[x];
17. j ← j+1;                  //move to next group
```

```
CLUSTERING (G)

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

5 CONCLUSION

In this paper we consider table annotation approach to automatically construct an annotation wrapper for annotating the search result records retrieved from any given database. The annotation approach is capable of generating high quality annotation. As we have problem with alignment, annotation approach helps to achieve good results. Many WDBs use a table to organize the returned SRRs. In the table, each row represents an SRR. Usually, the data units of the same concepts are well aligned. This special feature of the table layout can be utilized to annotate the SRRs. An annotation wrapper for the search site is automatically constructed and can be used to annotate new result pages from the same web database. Our experiments indicate that the proposed approach is good and effective.

6 FUTURE SCOPE

The result occurred shows that the data units and the presentation style are the most important features in alignment method. Then, we apply our annotation method to determine the success rate of annotator. We considered table annotator method in order to obtain good and effective results. This work can be extended further by using the other basic annotators like query based annotator, schema value annotator, in text prefix/suffix annotator, and common knowledge annotator to obtain more robust wrapper.

REFERENCES

[1] J. Madhavan, D. Ko, L. Lot, V. Ganapathy, A. Rasmussen, and A.Y. Halevy, "Google's Deep Web Crawl," Proc. VLDB Endowment, vol. 1, no. 2, pp. 1241-1252, 2008.

[2] Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Menig Celement Yu,"Annotating Search Results from Web Databases," IEEE Transactions on Knowledge And Data Engineering Vol. 25 No.3 March 2013

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

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

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

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

[7] Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng(2013)' *Annotating Search Results from Wel7 b Databases'*, IEEE Transactions on knowledge and data Engineering, vol. 25, no. 3.

[8] 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.

[9] J. Zhu, Z. Nie, J. Wen, B. Zhang, and W.-Y. Ma (2006) '*Simultaneous Record Detection and Attribute Labeling in Web Data Extraction*,'Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining.

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

[12] J. Wang and F.H. Lochovsky (2003)'*Data Extraction and Label Assignment for Web Databases'* Proc. 12th Int'l Conf. World Wide Web (WWW).