ISSN No. 0976-5697

Volume 5, No. 1, Jan-Feb 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Semantic Annotation for Search Results Based on Heuristics

Priyavarshini.P[1], Theetchenya.S[2]
[1,2]Deparment of Computer Science and Engineering
Sona College of Technology Salem, Tamil Nadu, India
[1]priyavarshini90@gmail.com
[2]theetchenya@gmail.com

Abstract- Search Engines generate a set of long flat list of information in response to user query from online databases. Information extraction and integration is most for many applications such as online comparison shopping and web data collection. Result pages generated dynamically by a search engines comes from multiple web databases. The structured data extracted from these pages with annotations are valuable for human browsing and machine processing which makes information retrieval system efficient. We propose a new EAAW System to automatically Extract, Align and Annotate the data items using heuristic information. Finally, Wrapper generation for the site is automatically constructed.

Keywords- Data alignment, Data annotation, Web database, Wrapper generation.

I. INTRODUCTION

Search engines are very important tools for people to reach the vast information on the World Wide Web. Recent studies indicate that Web searching, behind email, is the second most popular activities on the Internet. Surveys indicate that there are hundreds of thousands of search engines on the Web. Not only Web users interact with search engines, many Web applications also need to interact with search engines. For example, metasearch engines utilize existing search engines to perform search and need to extract the search results from the result pages returned by the search engines used. As another example, deep web crawling is to crawl documents or records from search engines and it too needs to extract the search results from the result pages returned by search engines. Deep web primarily contains some dynamic pages returned by underlying databases. We refer the accessed online databases on the web as “Web Data Bases (WDB)”.Compared with Surface Web; Deep Web contains more abundant, valuable and professional information, which cannot be indexed by traditional search engines. But the characteristics of large scale, heterogeneity, autonomy and distributed, makes how to make it effective advantage of Web information challenging. Thinking that structural information of database model is lost through searching, and current search results are for only human browsing, so applications cannot understand this information extracted from result pages. For a higher value these data must be understandable and for machine processing. Semantic annotation aims to add corresponding information to Deep Web search results, which enables computer understand and process these data.

The result pages will have multiple Search Result Records (SRR) or chunks. Each SRR will consists of text items and each text item will have data items (or instances) belonging particular entity. All data items are not encoded with meaningful labels for machine processing.

For search engines that have a Web service interface like Google and Amazon.com, automated tools may be used to extract their SRRs because the result formats are clearly described in the WSDL file of the Web services. However, our investigation indicates that very few search engines have Web services interfaces currently. One reason may be that Web services are designed to support B2B applications while most search engines are B2C applications. Therefore, we need to deal with search engines with no Web service interfaces and extract results that are presented in HTML files. This paper will address the problem on how to automatically annotate the data items. According to our work annotating data items refers to assigning significant labels to them .In this paper, we propose a novel annotation technique to automatically construct annotations for any given web search results. Given result pages of a web databases, we first extract the SRRs from these pages. Then, extract the data items from those records.

The extracted data items are clustered into respective semantic groups based on similarity measures which have been proposed [1] and nested data structured are also processed. The data items are then annotated. Finally with the annotated data items, an annotation wrapper is constructed for the web database which can be used to annotate new SRRs retrieved from the web database in response to new queries. A new semantic annotation method is proposed which analysis the characteristic features of the interface page and result page to summarize some heuristic information. This method uses this heuristic information in turn to analyze the data to be annotated, for identifying a semantic vocabulary for each data unit.

Our method proposes new heuristic information, which is the adjacent position characteristic of attributes and word count of attribute value. This new point will contribute to the annotation completeness, thus recall of the annotation will be improved. Compared with Ontology-based annotation method, our method has a better performance. Instead of flat list, it groups the SRRs into clusters, and annotates with a representative word to each cluster. Then, these labeled clusters of results are presented to users. Users can benefit from labeled clusters because the size of information presented is significantly reduced and enables fast navigation of Web page information retrieval.

II. RELATED WORKS

Annotating search results refers to extracting structured data from deep web were cluster based shifting technique is used for clustering same semantic data (alignment) by handling all types of relations between text units and data units and then annotated using Multi-annotator Approach where each annotator exploits a special feature [1]. Current annotation methods are mainly integrated interface schema based, local interface schema based and others. Multi-annotator approach [1] where multiple annotators are used and where annotators are fairly independent from each other since each exhibits a special feature. The Annotators [1] used are: Table Based Annotator, Query Based Annotator, Schema Value Based Annotator, Frequency Based Annotator, In-text Based Annotator, and Common Knowledge Based Annotator. A special feature of this method is that, when annotating the search results retrieved from Web database, it utilizes both the Local Interface Schema (LIS) of a local web database and Integrated Interface Schema (IIS) of multiple web databases in the same domain.IIS has helped to alleviate the local interface schema inadequacy problem and the inconsistent label problem. In order to combine different annotators, we employ a simple probabilistic method to combine different annotators. Combined probability is used to combine the annotators. One advantage of this model is its flexibility in the sense that when an existing annotator can be modified and new annotators can be added.

Data extraction and label assignment [3] is the most similar to our work. In this approach, alignment is purely based on HTML tags and utilizes different search interface s of WDBs for annotation. It first uses HTML tags to align data units by filling them into a table through regular expression based on data tree algorithm. Then, it employs four heuristics to select a label for each aligned table column. The data alignment is purely based on HTML tags and handles only two types of relations. Only Local Interface Schema is used to annotate the data units.

To enable fully automatic annotation, the result pages have to be automatically obtained and the SRRs need to be automatically extracted. In a metasearch context result pages are retrieved by queries submitted by users (some reformatting may be needed when the queries are dispatched to individual WDBs). Simultaneous Record Detection and Attribute Labeling in Web Data Extraction [6] perform attribute extraction and labeling simultaneously. However, the label set is predefined and contains only a small number of values.

Automatic annotation approach [5] exploits the spatial co-ordinate distance between data- value and label in the graphical rendering of web page, to find out an optimal label for each data-value. But this method is just apt to the situation where all the labels of query-result exist in the query-result page.

In ODE system [4], the ontology for a domain is constructed by matching the query interfaces and the query result pages among different websites. Then, the ontology is used to do the data extraction. For query result section identification, ODE finds a sub tree, which has the maximum correlation with the ontology, in the HTML tag tree. For data value alignment and label assignment, ODE uses a maximum entropy model. Context, tag structure and visual information are used as features for the maximum entropy model. The ontology assisted data extraction method is fully automatic and can avoid many of the problems that exist in most current automatic data extraction methods.

Search result clustering helps users to browse the search results and locate what they are looking for. In the search result clustering, the label selection which annotates a meaningful phrase for each cluster becomes the most fundamental issue. In Language Label Model [9] present a new method of using the language modeling approach over Dmoz for label selection, namely label language model.

ViDE system [8] focuses on structured web data extraction problem, including data record extraction and data item extraction. This approach includes four steps: Visual Block Tree building, data record extraction, data item extraction and visual wrapper generation. Visual block tree building is to build the visual block tree for a given sample deep page using VIPS algorithm. With the visual block tree, data record extraction and data item extraction are carried out based on visual features. Visual wrapper generation is to generate the wrappers that can improve the efficiency of both data record extraction and data item extraction. ViDE can only process deep Web pages containing one data region, while there is significant number of multi-data region deep web pages which cannot be processed.

CTVS system [11] extracts QRRs from the result page. CTVS employs two steps for this task. The first step identifies and segments the QRRs. The second step aligns the data values among the QRRs. A novel alignment method is proposed in which the alignment is performed in three consecutive steps: Pairwise alignment, holistic alignment, and nested structure processing. This method shows accurate data extraction and suffers from limitations.

Idea Management Systems [12] are a tool for collecting ideas for innovation from large communities. One of the problems of those systems is difficult to depict the distinctive features of ideas in a rapid manner and use them for judgment of proposed innovations. Our research aims to solve this problem by introducing annotation of idea with a domain independent taxonomy that describes various characteristics of ideas. The findings of our study show that such annotations can be successfully transformed into new metrics that allow comparison of ideas with similar successfulness as metrics already used in Idea Management Systems but in greater detail. Based on the survey analysis of these result the annotation system has been devised.

III. PROPOSED SYSTEM

A. Architectural Design:

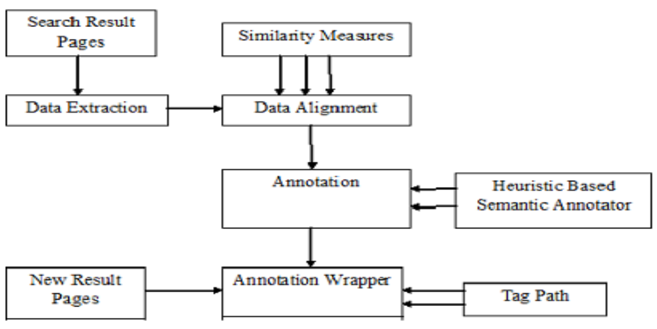

Figure 1 Architectural Design

B. Overview:

a. Extraction:

To enable annotation, search results have to be extracted out from the result pages returned by search sites. The major issue is to extract SRRs correctly while discarding irrelevant information such as advertisements' and sponsored links from each result page. ViNTs system [2] is used to extract the SRRs from search engine returned result pages.

Each SRR is stored in a tree structure with a single root and each node in the tree corresponds to a HTML tag or a piece of text in the original page. With this structure, it becomes easy to locate each node in the HTML page. Each node in the tag structure is either a tag node or a text node. Tag node corresponds to an HTML tag surrounded by "<" and ">".Text nodes are visible elements and data items are located in the text nodes. Since our annotation is at the data item level, data items must be identified from text nodes. Depending on the number of data items a text node contains four relationships are determined:

a) On-to-One Relationship: In this type each text node contains exactly one data item.
b) One-to- Many: In this type multiple data items are encoded in one text node.
c) Many-to-One Relationship: In this type, multiple text nodes together form a data item.
d) One-to-Nothing Relationship: In this case, the text nodes belonging to this category are not part of any data unit inside SRRs.

b. Alignment:

The extracted data has to be aligned for annotation. The data item alignment process will be based on the similarity features and trained weights are added to these similarity features using Genetic algorithm [7].The feature weights are also trained. The similarity features used for alignment of data items are: Content, Type, Format, Path, In-text, Semantic measures. Based on these calculated similarities the data items are aligned into columns using data item alignment algorithm. The clustering threshold used for clustering the data items are trained and calculated. Adjacency similarity is calculated for the prefix and suffix of the data items using jaccord similarity measure only because it reduces the time complexity.

a) Semantic similarity is a measure that derives the synonyms set from WorldNet database. The semantic similarity equation is shown as Equation
$SemSim(d_1,d_2) = 1 \text{ if } synset(d_1) = synset(d_2)$ (1)
$SemSim(d_1,d_2) = 0 \text{ if } synset(d_1) \neq synset(d_2)$ (2)
b) Content similarity is calculated using cosine similarity between the term frequency vectors of d_1 and d_2 and the function is
$SimC(d_1,d_2) = \frac{Vd1.vd2}{|vd1|*|vd2|}$ (3)
c) Presentation style similarity is calculated using style feature scores over all six presentation style features between d_1 ands,d_2
$SimP(d_1,d_2) = \sum_{i=1}^{6} \frac{FSi}{6}$ (4)
The features considered are bold, italic,, font color, size, decoration, face.
d) Data Type similarity is determined by the common sequence of the component data types between two data items.
$SimD(d_1,d_2) = LCS(d_1,d_2)$ (5)
e) Tag path similarity is determined using edit distance measure measure . Let p1 and p2 be the tag paths of d_1 and d_2 , the tag path similarity between d_1 and d_2 is
$SimT(d_1,d_2) = 1-EDT(p_1,p_2)$ (6)
f) Prefix Similarity and suffix similarity are also calculated using cosine similarity only due to time complexity problem.

Total similarity is determined using

$$sim(d_1,d_2) = w_1 *SemSim(d_1,d_2) + w_{2*} SimC(d_1,d_2) + w_{3*} SimP(d_1,d_2) + w_4 * SimD(d_1,d_2) + w_5*SimT(d_1,d_2) + w_6*Simpre(d_1,d_2) + w_7*Simsuf(d_1,d_2)$$

```
1.  ALIGN(SRRs)
2.  j←1;
3.  while true
    //create alignment groups
4.  For i←1 to number of SRRs
          G_J←SRR[i][j];
5.  If G_j is empty
6.  Exit;
7.  V←CLUSTERING(G);
8.  If |V|>1
9.  S←□;
10. For x←1to number of SRRs
11.    For y←j+1 to SRR[i].length
12.       S←SRRs[x][y];
    //cluster into groups
13. V[c] = _{k=1to |v|} min (sim(V[k],S));
14. for k←1 to |V|
15.    foreach SRR[x][j] in V[k]
16.       Insert NIL at position j in SRR[x]
17.s j←j+1;
GROUPING
     1.  V←all data units in G
     2.  While |V|>1
     3.  best←0;
     4.  L←NIL; R←NIL;
     5.  Foreach A in V
     6.     Foreach 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 □ R to V;
     15. Else break loop;
     16. Return V;
```

c. Annotation:

Interface Schema (IS) and Result Schema (RS) reflect partial schema structure of backend databases. And interface page and result page are direct to users. So, the analysis of IS and RS is the main way to get Web database schema information. IS is the portal of Web database. It is a set of attributes for querying and can be seen as a view built on the corresponding Web database. For a given web site, all data records have a fixed schema. Each data record consists of some attributes fields composed of attribute names and attribute values. These attribute names form RS of the Web database. In a specific area, semantic word set can be seen finite. It is defined as follows:

$W= \{w_1, w_2, w_3…w_n\}$.

Among, w_j denotes a semantic vocabulary. For a specific domain, we collect many web sites. From some interface pages and result pages of each site, we collect enough many vocabularies representing data concept of this domain. Based on domain knowledge, it needs to make vocabulary supplement and amendment for consistency of semantic description in different sites.

Semantic annotation is to find an explicit semantic vocabulary for every data unit in result pages, for making these data understandable and process able for computers. It's defined as follows: Suppose that there is a vocabulary set, $W=\{w_1,w_2,w_3,…..w_n\}$, group of attribute values to be annotated, $V=\{ v_1,v_2,v_3,…..v_n\}$. Semantic annotation is that, for each $v_i \in V$, we can find an appropriate w_j, which can relatively accurately describe the semantic of v_i. Finally it constructs a set of mapping pairs like $\{(vi, w_j) \backslash v_i \in v, w_j \in W$, w_j is the semantic instruction of $v_i\}$. Heuristics used are as follows:

a) Prefix Text Information:

Through the comparison of two-two strings in the j^{th} data group, if there exits prefix text in this group we cut it from each attribute value. It's not apt to the case that all data units in a same column have exactly identical content. Then, we select the most matching vocabulary with this prefix text from annotation vocabulary set to annotate this column.

b) Query Keywords Information:

If the query keywords submitted in interface page, will appear in the same data group across all SRRs. For j^{th} data group, we make a string matching between each r^{kj} and query keyword. If the keyword is a substring of each r^{kj} ,it uses the attribute name corresponding to query keyword in interface page to annotate this group.

c) Predefined Attribute Information:

In "Advanced Search", there are often some attributes defined in the form of "selection_list", "radio" or "check" components, which predefine an attribute value list. Therefore, we need to analyze IS to find out these attributes and their corresponding predefined value lists, such as < $Attribute_i$,{dv1,dv2,….,dvk}>,{dv1,dv2,…..,dvk} denotes a predefined value lists of $Attribute_{i.}$. For j^{th} data group , we check whether each r_{kj} (k=1,2,3….,n) is contained in a $Attribute_i$'s predefined value list, if so it uses $Attribute_i$ to annotate this group.

d) Special Pattern Information:

If price, email-id, date of birth, isbn- regular expression match, then that attribute is used to annotate. It is easy to build all possible regular expression for these attributes. For j^{th} data group , we check whether all the r_{kj} (k=1,2,….,n) are matched with a same regular expression. If so, it selects the corresponding attribute to annotate this group.

e) Domain Knowledge Information:

If some attribute values often contain some featured phrases or vocabulary. According to domain knowledge, from annotation vocabulary set those data items can be annotated. If there two data items, \$34.50 and \$20.60 which are annotated with "price". According to domain knowledge, the larger number represents "original price" and the smaller number represents price "discount price".

f) Adjacent position characteristic and word count:

If a data item judged as "title" and adjacent data item is not found a label. Besides, if no data item is annotated with "Author". Then, "author" can be used to annotate. If an attribute value is greater than a certain number it can be annotated as description. Second is, the word count of some attribute value is often over a certain number. We call this number "Word Threshold". Such as "Book Description " ,whose containing word count must be relatively more than general attribute. As for "Word Threshold ", we can get it through training. Therefore, if all attribute values of a data group are beyond this threshold, it can use "Book Description" to annotate it.

Based on this heuristic information the data units are annotated. Input file will be an XML file containing the extracted and aligned data from a dynamically generated result page. Output file will contain data units with semantic labels, the corresponding annotation file in XML. Then, the following steps are followed for annotating:

(a). Annotation vocabularies for a specified domain are collected as a semantic annotation vocabulary set.
(b). Analyze the attributes and the predefined value list and make a mapping pair.
(c). After analyzing label the data units.

```
Heuristic Based Annotation Algorithm:

Annotate ()

//AM= Annotation Mapping QCL=Query Condition List

//PVLM=Predefined Value List Mapping FPML=Featured Phrase Mapping List

1.  for every d_j group
2.      If no semantic label
3.        AM←CheckIF_Prefix_Text();
4.        AM←CheckIF_Suffix_Text();
5.        AM←CheckIF_Query_Word(QCL);
6.        AM←CheckIF_Preval_List(PVLM);
7.        AM←CheckIF_Special_Attribute();
8.        AM←CheckIF_Feat_Phrase(FPML);
9.        AM←CheckIF_Adjacent()
10.       AM←CheckIF_wordcount();
11.     End If
12. End for
    //Return the semantic word.
13. Return AM
```

d. Annotation Wrapper:

For wrapper generation the method used in [1] is used which utilizes the tag path structure alone. Each annotated column corresponds to an attribute in SRRs. The annotation wrapper is a description of the annotation rules for all the attributes on the result page. After the data items are annotated, they are organized based on the order of its data items in the original SRRs.

To use the wrapper to annotate a new result page, for each data item in an SRR, the annotation rules are applied on it one by one based on the order they appear in the wrapper .If this data item has the same prefix and suffix as specified in the rule, the rule is matched and the item is labeled with the given label in the rule. If the separators are specified, they are used to split the item, and label is assigned to the item. The following steps are used to generate wrappers:

a) Using the annotated data items annotation wrapper can be generated for the Web databases so that the new SRRs retrieved from the same web database can be annotated.
b) Annotation Wrappers use only tag path.
c) Annotation rule for each attribute consists of 5 components expressed as :attribute$_i$=<label$_i$, prefix$_i$, suffix$_i$, seperators$_i$, unitindex$_i$>

IV. CONCLUSION

Assigning semantic labels to the data units extracted from result pages in a challenging task. In this paper, by analyzing the features of the interface page and result page, we proposed a heuristic information based annotation method. Automatic data alignment problem is also addressed. Our method is based on data clustering algorithm utilizing richer yet automatically obtainable features.

V. REFERENCE

[1]. Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng, Member, IEEE, and Clement Yu, Senior Member, Annotating Search Results from Web Databases, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 3, MARCH 2013.

[2]. 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 (WWW), 2005.

[3]. J. Y. Wang and F. H. Lochovsky. Data extraction and label assignment for web databases. In Proceedings of the 12th International Conference on World Wide Web, pages 187-196, 2003.

[4]. 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.

[5]. L. Arlotta, V. Crescenzi, G. Mecca and P. Merialdo. Automatic annotation of data extracted from large Web sites. In Proceedings of the 6th International Workshop on Web and Databases. San Diego, pages 7-12, 2003.

[6]. J. Zhu, Z. Nie, J. Wen, B. Zhang, and W.-Y. Ma, “Simultaneous Record Detection and Attribute Labeling in Web Data Extraction,”Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, 2006.

[7]. D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, 1989.

[8]. Wei Liu, Xiaofeng Meng, Member, IEEE, and Weiyi Meng, Member, IEEE ViDE: A Vision-Based Approach for Deep Web Data Extraction IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 3, MARCH 2010

[9]. Yeha Lee ,Seung-Hoon Na ,Jong –Hyeok Lee Search Result Clustering Using Label Language Model ,Open directory project, http://www.dmoz.com.

[10]. 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.

[11]. Weifeng Su,Jiying Wang,Frederick H.Lochovsky,Member IEEE Computer Society, and Yi Liu, Combining Tag and Value Similarity for Data Extraction and Alignment, , IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 3, MARCH 2013.

[12]. Adam Westerski, Theodore Dalamagas, Carlos A.Iglesias ,Classifying and comparing community innovation in Idea Management Systems, In Elsevier Journal.