www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11422-11425

Web Database Annotation for Fast and Accurate Retrieval

Kiran Dhumale, Swati Chavanm, Monali Kulkarni, Sachin Landge

(Department of Computer Engineering, DYPCOE Savitribai Phule University of Pune, Maharashtra, India)

Miss Disha Tiwari

(Asst Prof Department of Computer Engineering, DYPCOE Savitribai Phule University of Pune, MH, India)

I. ABSTRACT

Documents on the web exist in digital format, people spend large amount of time on searching using web browsers for finding useful information. The results returned by search engines are in the form of web pages that contains results obtained from different web databases. These results can be used further in many applications such as data collection, comparison of prices and many more but to make these applications successful the search results should be machine processable. so to make them machine processable, it is important that the result pages are annotated in a meaningful manner. The process of annotating has to consider groups of data and obtain final annotation after aggregating them. Annotation can be done for the Web, java, pdf files, text files, xps, mobile, image, multimedia etc. In Information retrieval for decision support and integrating an annotation database can be founded on the parameters such as document, user and time. Automatic extraction of the data from querying result pages is very much important for different applications, such as data integration, Meta querying cooperates with multiple web databases.

KEYWORDS: Automatic Annotation, Annotation Wrapper, Data Mining, Data Extraction, Information Retrieval.

II. INTRODUCTION

The Internet gives very large amount of information which is usually formatted for different users, which makes it very difficult to extract relevant and accurate data from various sources. So, the availability of robust, flexible information retrieval systems that transforms the Web pages from web databases into user friendly dynamic pages such as a database, structured data becomes a need. Information retrieval includes the representation, storage, organization of, and access to information items such as documents, Web pages, online catalogs, structured and semi-structured records, multimedia objects. Databases are technologies for managing very large amount of data. Efficiency of searching and updating information is increased by alignment algorithm and annotation. Data alignment is aligning the data or arranging the data in such a way that data inside the same group have the same meaning and accessing in computer memory. Data annotation is the methodology for adding information to a document, a word or phrase, paragraph or the entire document. Data annotation enables fast retrieval of information in the deep web. Annotate the data units requires lots of human efforts. Thus, lack in scalability. To overcome this, automatic assigning of data units within the SRRs is required. An automatic annotation approach that first arrange all data into different groups i.e. inside the same group have same semantic. Then each group is annotated in different aspects and aggregated to predict a final label. Finally, wrapper is generated.

III. RELATED WORK

In recent years, web information extraction and annotation is an important research area. The system proposed in [3] explains that the traditional approach takes more time to annotate the database. It also requires manual efforts. Automatically assigning the meaningful labels has been introduced in [3]. Also three annotation phases are described : alignment phase, annotation phase and annotation wrapper generation phase. In data extraction from large websites [3] annotates data units with their closest labels on the result page. This approach proposed that they maintain all type of relationship between the text nodes and data units. The wrapper induction system is introduced in which mark the label data and also rely on human users. However, this system achieves higher extraction precision in the result. In addition, this system undergoes lesser scalability that does not fit in the applications mentioned by authors . A similar approach is based on ontology means, automatically extracts the data from web documents. Author S. Mukherjee, discussed a method to align the data units which maintains only one type of relationship i.e. one to one relationship in between data unit and text nodes. Also a domain dependent annotation process has been introduced. An ontology based system insightful to the data quality has been introduced in. In automatically building a wrapper has been presented. These methods are used only for the data extraction, but not for annotation. The various methods discussed by the authors assign the labels to the data from the web databases [2].

IV.DATA UNIT AND TEXT NODE RELATIONSHIPS

Data unit is a piece of text that semantically represents concept of real world entity. Data unit is totally different from text node where, text node is a sequence of text surrounded by pair of HTML tags. Text node is visible element on the web page and data unit located in the text nodes. Relationships between text node and data unit features are:

4.1 One-to-One Relationship: (referred as atomic text nodes). Text node containing only one data unit i.e. the text of this node contains the value of a single attribute. Each text node surrounded by the pair of HTML tags <A> and </A>.

4.2 One-to-Many Relationship: (referred as composite text nodes) A text node consists of multiple data units i.e. multiple data units are encodes into single text nodes.

4.3 Many-to-One Relationship: (referred as decorative tags) multiple text nodes are encoded into single data unit.

4.4 One-To-Nothing Relationship: (referred as template text nodes) Text nodes are not part of any data unit inside SRRs. This relationship for text nodes and data units are represents the relation in between them.

V. DATA UNIT AND TEXT NODE FEATURES

5.1 Features Shared By Data Units:

5.1.1 Data Content: To search information quickly data unit or text node of same concepts shares certain keywords.

5.1.2 Presentation Style: This feature describes how a data unit is displayed on the web page by using few styles are out face, font size, color, text decoration etc.

5.1.3 Data Type: These features are predefined characteristics that have their own meaning. Basically used data types are date, time, currency, integer, decimal etc.

5.1.4 Tag Path: Sequence of tags traversing from root to corresponding node in the tree.

5.1.5 Adjacency: Adjacency refers to the data units that are immediately before and after in the SRR.

VI. ALIGNMENT ALGORITHM

Alignment algorithm has following four steps.

Step 1: Merge text nodes*:* This step detects and removes decorative tags from each SRR to allow the text nodes corresponding to the same attribute merge into a single one.

Step 2: Align text nodes: After the merging aligns text nodes into different groups. So that same group has the same concepts.

Step 3: Split text nodes: In this step split the composite text nodes into separate data unit.

Step 4: Align data units: This is the last step for alignment in which separates each composite group into multiple aligned groups with each containing the data units of the same concept.

VII. ASSIGNING LABELS

For a WDB, its search interface often contains some attributes of the underlying data. We denote a LIS as S_i = {A_1, A_2,… A_k } where each A_j is an attribute. When a query is submitted against the search interface, the entities in the returned results also have a certain hidden schema, denoted as S_e = {a_1, a_2, .. a_n }, where each a_j (j = 1 . . . n) is an attribute to be discovered. The schema of the retrieved data and the LIS usually share a significant number of attributes [5]. This observation provides the basis for some of our basic annotators. If an attribute a_t in the search results has a matched attribute A_t in the LIS, all the data units identified with at can be labeled by the name of A_t. However, it is quite often that S_e is not entirely contained in S_i because some attributes of the underlying database are not suitable or needed for specifying query conditions as determined by the developer of the WDB, and these attributes would not be included in S_i. This phenomenon raises a problem called local interface schema inadequacy problem. Specifically, it is possible that a hidden attribute discovered in the search result schema Se does not have a matching attribute A_t in the LIS S_i. In this case, there will be no label in the search interface that can be assigned to the discovered data units of this attribute. Another potential problem associated with using LISs for annotation is the inconsistent label problem, i.e., different labels are assigned to semantically identical data units returned from different WDBs because different LISs may give different names to the same attribute. This can cause problem when using the annotated data collected from different WDBs, e.g., for data integration applications. In our approach, for each used domain, we use WISEIntegrator [4]) to build an IIS over multiple WDBs in that domain. The generated IIS combines all the attributes of the LISs. For matched attributes from different LISs, their values in the local interfaces (e.g., values in selection list) are combined as the values of the integrated global attribute [4]. Each global attribute has a unique global name and an attribute-mapping table is created to establish the mapping between the name of each LIS attribute and its corresponding name in the IIS. In this paper, for attribute A in an LIS, we use $_{gn}$(A) to denote the name of A's corresponding attribute (i.e., the global attribute) in the IIS. For each WDB in a given domain, our annotation method uses both the LIS of the WDB and the IIS of the domain to annotate the retrieved data from this WDB. Using IISs has two major advantages. First, it has the potential to increase the annotation recall. Since the IIS contains the attributes in all the LISs, it has a better chance that an attribute discovered from the returned results has a matching attribute in the IIS even though it has no matching attribute in the LIS. Second, when an annotator discovers a label for a group of data units, the label will be replaced by its corresponding global attribute name (if any) in the IIS by looking up the attribute-mapping table so that the data units of the same concept across different WDBs will have the same label.

VIII. ANNOTATION WRAPPER

Once the data units on a result page have been annotated, we use these annotated data units to construct an annotation wrapper for the WDB so that the new SRRs retrieved from the same WDB can be annotated using this wrapper quickly without reapplying the entire annotation process. We now describe our method for constructing such a wrapper below. Each annotated group of data units corresponds to an attribute in the SRRs. The annotation wrapper is a description of the annotation rules for all the attributes on the result page. After the data unit groups are annotated, they are organized based on the order of its data units in the original SRRs. Consider the i^{th} group G_i. Every SRR has a tag-node sequence that consists of only HTML tag names and texts. For each data unit in G_i, we scan the sequence both backward and forward to obtain the prefix and suffix of the data unit. The scan stops when an encountered unit is a valid data unit with a meaningful label assigned. Then, we compare the prefixes of all the data units in G_i to obtain the common prefix shared by these data units. Similarly, the common suffix is obtained by comparing all the suffixes of these data units.

IV. EXPERIMENTAL RESUTLS

We have experimental data from two domains: *Books and Mobiles* The performance of data alignment and annotation are presented in Table 1.

Domain		Book	Mobiles
Data Alignment Performance	Precision	96.10%	96.20%
	Recall	95.20%	96.40%
Annotation Performance	Precision	95.20%	96.30%
	Recall	95.30%	95.60%
Annotation With Wrapper	Precision	92.60%	92.50%
	Recall	91.30%	91.20%

Table 1 –Experimental results

As presented in Table 1, it is evident that more than 90% 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.

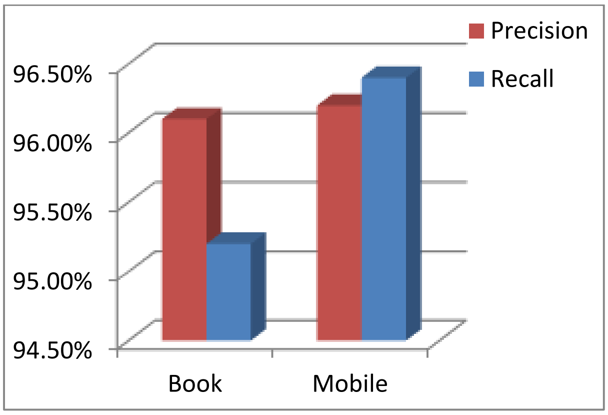

Figure 1 – Data Alignment Performance

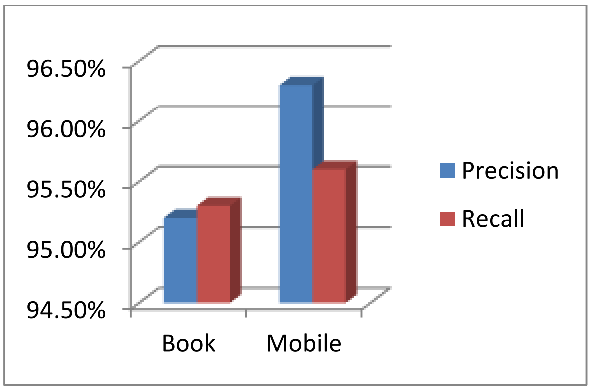

Figure 2 – Annotation Performance

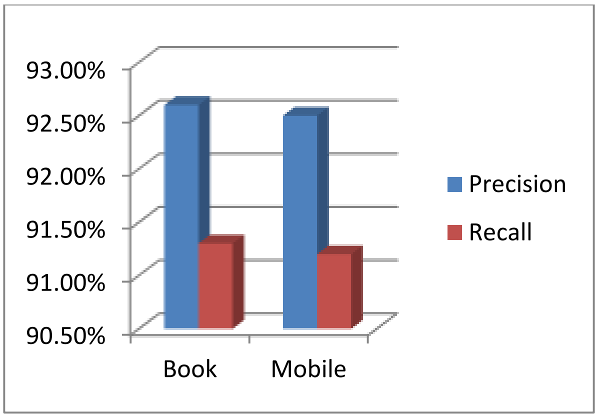

Figure 3– Annotation with wrapper Performance

X. CONCLUSION

Issues of relationship, scalability, wrapper induction, automatically data extraction are studied. Clustering approaches adopted in the literature are limited; hence there is scope for linking clustering based methods with data annotation approaches. Used search result records as a Database which will change accordingly. The automatic annotation approach considers several types of data unit and text node features and makes annotation scalable and automatic. Basically three phases used for automatic annotation in which aligns the data units into different groups, labels each group and construct an annotation wrapper. In this work not all data units are encoded with the meaningful labels. A new algorithm for data annotation in the web database would be proposed.

XI. REFERENCES

[1] Shilpa Jadhao, Prof. R. P. Kulkarni ,"Review of Semantic Web, Annotation Methods and Automatic Annotation for Web Search Results*", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 International Conference on Industrial Automation and Computing (ICIAC- 12-13th April 2014)*

[2] Miss.Priyanka P.Boraste,"A Survey on Data Annotation for the Web Databases ,"*IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727Volume 16, Issue 2, Ver. XI (Mar-Apr. 2014), PP 68-70*
www.iosrjournals.org

[3] Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng, Member, and Clement Yu, " Annotating Search Results from Web Databases", *IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 3, March 2013*

[4] H. He, W. Meng, C. Yu, and Z. Wu, "Automatic Integration of Web Search Interfaces with WISE-Integrator," *VLDB J., vol. 13, no. 3, pp. 256-273, Sept. 2004.*

[5] J. Wang, J. Wen, F. Lochovsky, and W. Ma, "Instance-Based Schema Matching for Web Databases by Domain-Specific Query Probing," *Proc. Very Large Databases (VLDB) Conf., 2004.*