CLUSTERING BASED ANNOTATION OF SEARCH RESULTS

Sini Thomas, PG Scholar(M.E C.S.E)

Guided by Mrs.J.Suganthi, Principal

Hindusthan College of Engineering & Technology, Othakkalmandapam, Pollachi Main Road

Coimbatore – 641 032, Tamil Nadu, India

Abstract

Deep web is a database based, i.e., for many search engines, data encoded in the returned result pages come from the underlying structured databases. 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). Unfortunately, the semantic labels of data units are often not provided in result pages. 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. Early applications require tremendous human efforts to annotate data units manually, which severely limit their scalability. In this project, 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 highly effective.

Keywords – Data Alignment, data annotation, web database, wrapper generation.

I. INTRODUCTION

Data Mining is the process of extracting hidden knowledge from large volumes of raw data. The knowledge must be new, not obvious, and one must be able to use it. Knowledge discovery differs from traditional information retrieval from databases. In traditional DBMS, database records are returned in response to a query; while in knowledge discovery, what is retrieved is not explicit in the database. Rather, it is implicit patterns. The Process of discovering such patterns is termed as data mining. There are two main reasons to use data mining like too much data and too little information and also there is a need to extract useful information from the data and to interpret the data.

Due to enormous volumes of data, human analysts with no special tools can no longer find useful information. However, Data mining can automate the process of finding relationships and patterns in raw data and results can be utilized in an automated decision support system or assessed by a human analyst. That is why the data mining is very useful, especially in science and business areas which need to analyze large amounts of data to discover trends in it. The data mining would be one of the valuable assets, if we know how to reveal valuable knowledge that is hidden in the raw data. The data mining is a tool to extract diamonds of knowledge from the historical data and can also predict the outcomes of future situations.

Data unit is a piece of text that semantically represents one concept of an entity. It corresponds to the value of a record under an attribute. It is different from a text node which refers to a sequence of text surrounded by a pair of HTML tags. In this research , we perform data unit level annotation. There is a high demand for collecting data of interest from multiple WDBs. For example, once a book comparison shopping system collects multiple result records from V different book sites, it needs to determine whether any two SRRs refer to the same book. The ISBNs can be compared to achieve this. If ISBNs are not available, their titles and authors could be compared. The system also needs to list the prices offered by each site. Thus, the system needs to know the semantic of each data unit. Unfortunately, the semantic labels of data units are often not provided in result pages. The semantic labels for the values of title, author, publisher, 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 (e.g., Deep web crawlers) for later analysis. Early applications require tremendous human efforts to annotate data units manually, which severely limit their scalability. In this research we consider how to automatically assign labels to the data units within the SRRs returned from WDBs.

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 main objective of the paper is large deep collection of web pages results with many search engines and encoded the data unit returned from the web databases with the label annotation with data unit that are returned from the web database with HTML form based results.

II. RELATED WORK

A. ViDE: A Vision-Based Approach for Deep Web Data Extraction

Number of Web databases has reached 25 million according to a recent survey. All the Web databases make up the deep Web (hidden Web or invisible Web). Often the retrieved information (query results) is enwrapped in Web pages in the form of data records. These special Web pages are generated dynamically and are hard to index by traditional crawler based search engines, such as Google and Yahoo. In this paper, we call this kind of special Web pages deep Web pages. Each data record on the deep Web pages corresponds to an object

Extracting structured data from deep Web 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 inherent limitations because they are Web-page-programming-language dependent. As the popular two-dimensional media, the contents on Web pages are always displayed regularly for users to browse. This motivates us to seek a different way for deep Web data extraction to overcome the limitations of previous works by utilizing some interesting common visual features on the deep Web pages. In this paper, a novel vision-based approach that is Web-page programming- language-independent is proposed. This approach primarily utilizes the visual features on the deep Web pages to implement deep Web data extraction, including data record extraction and data item extraction.

B. On Deep Annotation

Several approaches have been conceived (e.g. CREAM, MnM, or Mindswap) that deal with the manual and/or the semiautomatic creation of metadata from existing information. These approaches, however, as well as older ones that provide metadata, e.g. for search on digital libraries, build on the assumption that the information sources under consideration are static, e.g. given as static HTML pages or given as books in a library. Nowadays, however, a large percentage of Web pages are not static documents. On the contrary, the majority of Web pages are dynamic.2 For dynamic web pages (e.g. ones that are generated from the database that contains a catalogue of books) it does not seem to be useful to manually annotate every single page. Rather one wants to “annotate the database” in order to reuse it for one’s own Semantic Web purposes. For this objective, approaches have been conceived that allow for the construction of wrappers by explicit definition of HTML or XML queries or by learning such definitions from examples. Thus, it has been possible to manually create metadata for a set of structurally similar Web pages.

The wrapper approaches come with the advantage that they do not require cooperation by the owner of the database. However, their shortcoming is that the correct scraping of metadata is dependent to a large extent on data layout rather than on the structures underlying the data. While for many web sites, the assumption of non-cooperatively may remain valid, we assume that many web sites will in fact participate in the Semantic Web and will support the sharing of information. Such web sites may present their information as HTML pages for viewing by the user, but they may also be willing to describe the structure of their information on the very same web pages.

The success of the Semantic Web crucially depends on the easy creation, integration and use of semantic data. For this purpose, we consider an integration scenario that defies core assumptions of current metadata construction methods. We describe a framework of metadata creation when web pages are generated from a database and the database owner is cooperatively participating in the Semantic Web. This leads us to the definition of ontology mapping rules by manual semantic annotation and the usage of the mapping rules and of web services for semantic queries. In order to create metadata, the framework combines the presentation layer with the data description layer — in contrast to “conventional” annotation, which remains at the presentation layer. Therefore, we refer to the framework as deep annotation

3. Annotating Structured Data of the Deep Web

Currently, a large portion of the deep Web is database-based, i.e., data encoded in the returned result pages of many search engines come from the underlying structured databases (e.g., relational databases). Such type of search engines will be referred to as Web databases in this paper. A typical result page of a Web database consists of multiple search result records (SRRs) and each SRR corresponds to an entity. For example, each of the three SRRs contains information about a book. Usually, each SRR consists of multiple data units (or instances) like book title, author, publisher, price, etc. Frequently, not all data units are encoded with meaningful labels. For example, the First line of the First SRR is not labelled with “title" even though human users can recognize it easily. This paper addresses how to automatically annotate the data units in the SRRs returned by Web databases. In this paper we refer annotating data units as assigning meaningful labels to them.

Search sites that have Web services interfaces, it may be easier to annotate their SRRs because the semantic meanings of their data units are more clearly described in WSDL. However, our investigation indicates that very few search sites have Web services interfaces. One reason for this phenomenon may be that Web services are primarily designed to support B2B applications while most search sites are for B2C applications.

Holistic and multi-annotator approach to automatically constructing an annotation wrapper for any given Web database. Given a set of sample result pages of a Web database, we first extract the SRRs from these pages. Then the data units in all SRRs are aligned such that all data units in each aligned group semantically belong to the same attribute/concept. We then design different basic annotators to annotate data units in each aligned group holistically. The results of different basic annotators are combined to determine an appropriate label for each group of data units. Finally, with the annotated data units, 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.

III. SYSTEM IMPLEMENTATION

A. Data Alignment

Data alignment is based on the following similarities: Data unit similarity, Data content similarity, Presentation style similarity, Data type similarity, Tag path similarity, Adjacency similarity.

Data Unit Similarity

Data alignment is to put the data units of the same concept into one group so that they can be annotated holistically. Whether two data units belong to the same concept is determined by how similar they are based on the features,

$$\begin{aligned} Sim(d_1, d_2) = {} & w_1 * SimC(d_1, d_2) + w_2 * SimP(d_1, d_2) \\ & + w_3 * SimD(d_1, d_2) + w_4 * SimT(d_1, d_2) \\ & + w_5 * SimA(d_1, d_2). \end{aligned}$$

Data content similarity (SimC). It is the Cosine similarity between the term frequency vectors of d_1 and d_2:

$$SimC(d_1, d_2) = \frac{V_{d_1} \bullet V_{d_2}}{||V_{d_1}|| * ||V_{d_2}||},$$

where V_d is the frequency vector of the terms inside data unit d, $||V_d||$ is the length of V_d, and the numerator is the inner product of two vectors.

Presentation style similarity (SimP). It is the average of the style feature scores (FS) over all six presentation style features (F) between d_1 and d_2

$$SimP(d_1, d_2) = \sum_{i=1}^{6} FS_i/6,$$

where FS_i is the score of the ith style feature and it is defined by $FS_i = 1$ if $F_d^{1\,=}\,F_d^2$ and $FS_i = 0$ otherwise, and $F_d^{\,I}$ is the ith style feature of data unit d

Data type similarity (SimD). It is determined by the common sequence of the component data types between two data units. The longest common sequence (LCS) cannot be longer than the number of component data types in these two data units. Thus, let t_1 and t_2 be the sequences of the data types of d_1 and d_2, respectively, and TLen(t) represent the number of component types of data type t, the data type similarity between data units d_1 and d_2 is

$$SimD(d_1, d_2) = \frac{LCS(t_1, t_2)}{Max(Tlen(t_1), Tlen(t_2))}.$$

Tag path similarity (SimT). This is the edit distance (EDT) between the tag paths of two data units. The edit distance here refers to the number of insertions and deletions of tags needed to transform one tag path into the other. It can be seen that the maximum number of possible operations needed is the total number of tags in the two tag paths. Let p_1 and p_2 be the tag paths of d_1 and $d_{2,}$ respectively, and PLen(p) denote the number of tags in tag path p, the tag path similarity between d_1 and d_2 is

$$SimT(d_1, d_2) = 1 - \frac{EDT(p_1, p_2)}{PLen(p_1) + PLen(p_2)}.$$

Adjacency similarity (SimA). The adjacency similarity between two data units d1 and d2 is the average of the similarity between d_p^1 and d_p^2 and the similarity between d_s^1 and d_s^2, that is

$$SimA(d_1, d_2) = \left(Sim'(d_1^p, d_2^p) + Sim'(d_1^s, d_2^s)\right)/2.$$

B. Algorithm

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 (separated by decorative tags) to be merged into a single text node.

Step 2: Align text nodes. This step aligns text nodes into groups so that eventually each group contains the text nodes with the same concept (for atomic nodes) or the same set of concepts (for composite nodes).

Step 3: Split (composite) text nodes. This step aims to split the “values” in composite text nodes into individual data units. This step is carried out based on the text nodes in the same group holistically. A group whose “values” need to be split is called a composite group.

Step 4: Align data units. This step is to separate each composite group into multiple aligned groups with each containing the data units of the same concept.

C. Annotation Wrapper

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. 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 ith group G_i. Every SRR has a tag-node sequence like Fig. 1b 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.

IV PERFORMANCE EVALUATION

Precision and recall measures from information retrieval to evaluate the performance of our methods. For alignment, the precision is defined as the percentage of the correctly aligned data units over all the aligned units by the system; recall is the percentage of the data units that are correctly aligned by the system over all manually aligned data units by the expert.

A. Precision

Precision value is calculated is based on the retrieval of information at true positive prediction, false positive .In healthcare data precision is calculated the percentage of positive results returned that are relevant.

Precision =TP/ (TP+FP)

B. Recall

Recall value is calculated is based on the retrieval of information at true positive prediction, false negative. In healthcare data precision is calculated the percentage of positive results returned that are Recall in this context is also referred to as the True Positive Rate. Recall is the fraction of relevant instances that are retrieved,

Recall =TP/(TP+FN)

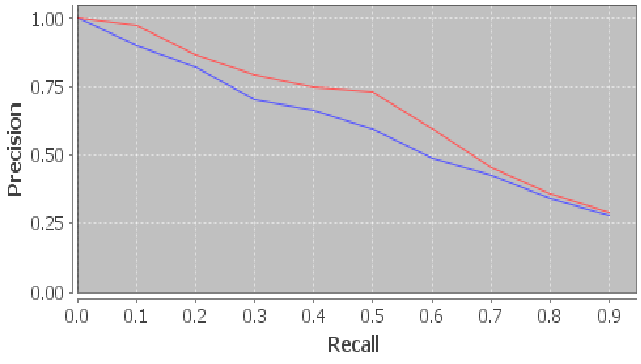

Figure 1. Precision vs Recall

V. CONCLUSION AND FUTURE ENHANCEMENT

Data annotation problem and proposed a multiannotator approach to automatically constructing an annotation wrapper for annotating the search result records retrieved from any given web database. This approach consists of six basic annotators and a probabilistic method to combine the basic annotators. Each of these annotators exploits one type of features for annotation and our experimental results show that each of the annotators is useful and they together are capable of generating high- quality annotation. A special feature of our method is that, when annotating the results retrieved from a web database, it utilizes both the LIS of the web database and the IIS of multiple web databases in the same domain. Accurate alignment is critical to achieving holistic and accurate annotation. Our method is a clustering based shifting method utilizing richer yet automatically obtainable features. This method is capable of handling a variety of relationships between HTML text nodes and data units, including one-to-one, one-to-many, many-to-one, and one-to-nothing.

We need to enhance our method to split composite text node when there are no explicit separators. We would also like to try using different machine learning techniques and using more sample pages from each training site to obtain the feature weights so that we can identify the best technique to the data alignment problem. To derive the feature weights SVM technique can be used.

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] P. Chan and S. Stolfo, “Experiments on Multistrategy Learning by Meta-Learning,” Proc. Second Int’l Conf. Information and Knowledge Management (CIKM), 1993.

[4] W. Bruce Croft, “Combining Approaches for Information Retrieval,” Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval, Kluwer Academic, 2000.

[5] V. Crescenzi, G. Mecca, and P. Merialdo, “RoadRUNNER: Towards Automatic Data Extraction from Large Web Sites,” Proc. Very Large Data Bases (VLDB) Conf., 2001.

[6] S. Dill et al., “SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation,” Proc. 12th Int’l Conf. World Wide Web (WWW) Conf., 2003.

[7] H. Elmeleegy, J. Madhavan, and A. Halevy, “Harvesting Relational Tables from Lists on the Web,” Proc. Very Large atabases (VLDB) Conf., 2009.

[8] 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.

[9] D. Freitag, “Multistrategy Learning for Information Extraction,” Proc. 15th Int’l Conf. Machine Learning (ICML), 1998.

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

[11] S. Handschuh, S. Staab, and R. Volz, “On Deep Annotation,” Proc. 12th Int’l Conf. World Wide Web (WWW), 2003.

[12] S. Handschuh and S. Staab, “Authoring and Annotation of Web Pages in CREAM,” Proc. 11th Int’l Conf. World Wide Web (WWW), 2003.

[13] B. He and K. Chang, “Statistical Schema Matching Across Web Query Interfaces,” Proc. SIGMOD Int’l Conf. Management of Data, 2003.

[14] 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.

[15] 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.

[16] J. Heflin and J. Hendler, “Searching the Web with SHOE,” Proc. AAAI Workshop, 2000.

[17] L. Kaufman and P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990.

[18] N. Krushmerick, D. Weld, and R. Doorenbos, “Wrapper Induction for Information Extraction,” Proc. Int’l Joint Conf. Artificial Intelligence (IJCAI), 1997.

[19] J. Lee, “Analyses of Multiple Evidence Combination,” Proc. 20th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval, 1997.

[20] L. Liu, C. Pu, and W. Han, “XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources,” Proc. IEEE 16th Int’l Conf. Data Eng. (ICDE), 2001.

[21] 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.