Web Data Extraction and Alignment

M. Jude Victor[1], D. John Aravindhar[2], V. Dheepa[3]

[1,2,3] School of Computing Sciences, Hindustan University
Vellore, Tamilnadu, India

judevctr@gmail.com
jaravindhar@hindustanuniv.ac.in
vdeepa@hindustanuniv.ac.in

Abstract: ***Web databases generate query result pages based on a user's query. Automatically extracting the data from these query result pages is very important for many applications, such as data integration, which need to cooperate with multiple web databases. We present a novel data extraction and alignment method called CTVS that combines both tag and value similarity. CTVS automatically extracts data from query result pages by first identifying and segmenting the query result records (QRRs) in the query result pages and then aligning the segmented QRRs into a table, in which the data values from the same attribute are put into the same column. We also design a new record alignment algorithm that aligns the attributes in a record, first pair wise and then holistically, by combining the tag and data value similarity information. Experimental results show that CTVS achieves high precision and outperforms existing state-of-the-art data extraction methods.***

Keywords: Data Extraction, Automatic Wrapper Generation, Data Record Alignment, Information Integration

1. Introduction

Online databases, called web databases, comprise the deep web tag and value. Compared with WebPages in the surface web, which can be accessed by a unique URL, pages in the deep web are dynamically generated in response to a user query submitted through the query interface of a web database. Upon receiving a user's query, a web database returns the relevant data, either structured or semi structured, encoded in HTML pages. Many web applications, such as met querying, data integration and comparison shopping, need the data from multiple web databases. For these applications to further utilize the data embedded in HTML pages, automatic data extraction is necessary. Only when the data are extracted and organized in a structured manner, such as tables, can they be compared and aggregated.

Hence, accurate data extraction is vital for these applications to perform correctly. This paper focuses on the problem of automatically extracting data records that are encoded in the query result pages generated by web databases. In general, a query result page contains not only the actual data, but also other information, such as navigational panels, advertisements, comments, information about hosting sites, and so on. The goal of web database data extraction is to remove any irrelevant information from the query result page, extract the query result records (referred to as QRRs in this paper) from the page, and align the extracted QRRs into a table such that the data values1 belonging to the same attribute are placed into the same table column.

2. Literature Survey

2.1 A Flexible Learning System for Wrapping Tables and Lists in HTML Documents

A program that makes an existing website look like a database is called a wrapper. Wrapper learning is the problem of learning website wrappers from examples. We present a wrapper-learning system called WL2 that can exploit several deferent representations of a document. Examples of such deferent representations include DOM-level and token-level representations, as well as two-dimensional geometric views of the rendered page (for tabular data) and representations of the visual appearance of text as it will be rendered. Additionally, the learning system is modular, and can be easily adapted to new domains and tasks. The learning system described is part of an industrial-strength" wrapper management system that is in active use at WhizBang Labs. Controlled experiments show that the learner has broader coverage and a faster learning rate than earlier wrapper-learning systems.

2.2 A Structured Wrapper Induction System for Extracting Information from Semi-Structured Documents

An extensible architecture which allows wrapper-learning systems to be easily constructed and tuned. In this architecture the bias of the wrapper-learning system is encoded as an ordered set of "builders", each associated with some restricted extraction language L. To implement a new builder it is only necessary to implement a small set of core operations for L. Builders can also be constructed by combining other builders. A single master learning algorithm which invokes the builders handles most of the real work of learning. The learning system described here is fully implemented, and is part of an "industrial-strength" wrapper-learning system which has been used to extract job postings from more than 500 sites.

2.3 Extracting Structured Data from Web Pages

Many web sites contain large sets of pages generated using a common template or layout. For example, Amazon lays out the author, title, comments, etc. in the same way in all its book pages. The values used to generate the pages (e.g., the author, title,) typically come from a database. In this paper, we study the problem of automatically extracting the database values from such template generated web pages without any learning examples or other similar human input. We formally define a template, and propose a model that describes how values are encoded into pages using a template. We present an algorithm that takes, as input, a set of template-generated pages, deduces the unknown template used to generate the pages, and extracts, as output, the values encoded in the pages. Experimental evaluation on a large number of real input page collections indicates that our algorithm correctly extracts data in most cases.

2.4 Data Extraction and Label Assignment for Web Databases

Many tools have been developed to help users query, extract and integrate data from web pages generated dynamically from databases, i.e., from the Hidden Web. A key prerequisite for such tools is to obtain the schema of the attributes of the retrieved data. In this paper, we describe a system called, DeLa, which reconstructs (part of) a "hidden" back-end web database. It does this by sending queries through HTML forms, automatically generating regular expression wrappers to extract data objects from the result pages and restoring the retrieved data into an annotated (labeled) table.

3. Existing System

Many web sites contain a large collection of “structured” web pages. These pages encode data from an underlying structured source, and are typically generated dynamically. An example of such a collection is the set of book pages in Amazon. There are two important characteristics of such a collection: first, all the pages in the collection contain structured data conforming to a common schema; second, the pages are generated using a common template. Our goal is to automatically extract structured data from a collection of pages described above, without any human input like manually generated rules or training sets. Extracting structured data gives us greater querying power over the data and is useful in information integration systems.

4. Proposed System

A novel data extraction method, CTVS, to automatically extract QRRs from a query result page. CTVS employs two steps for this task. The first step identifies and segments the QRRs. We improve on existing techniques by allowing the QRRs in a data region to be noncontiguous. The second step aligns the data values among the QRRs. Although CTVS has been shown to be an accurate data extraction method, it still suffers from some limitations. First, it requires at least two QRRs in the query result page. Second, any optional attribute that appears as the start node in a data region will be treated as auxiliary information. Third, similar to other related works, CTVS mainly depends on tag structures to discover data values.

5. Module Description

5.1 System Architecture

The following architecture is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation User request;

1. User Interface Design
2. Tag Tree Construction module
3. Data Region Identification module
4. Record Segmentation module
5. Query Result Section Identification module

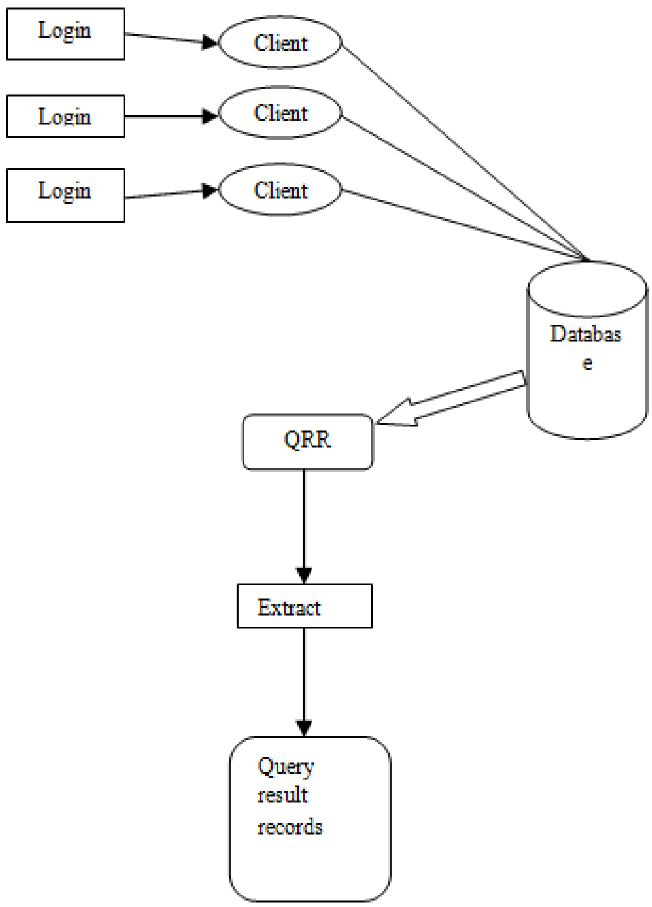

Figure 1: Describes the System Architecture process

5.2 User Interface Design

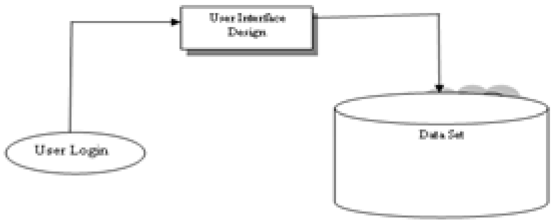

Figure 2: User to Design

In this module user has to create an account for only allowing right persons to access the resources. All the details will be stored in database which is placed in server.

If he entered correct user name and password then he will be able to access the cloud. Logging in is usually used to enter a specific page, which trespassers cannot see. Once the user is logged in, the login token may be used to track what actions the user has taken while connected to the site.

5.3 Tag Tree Construction Module

First constructs a tag tree for the page rooted in the <HTML> tag. Each node represents a tag in the HTML page and its children are tags enclosed inside it. Each internal node n of the tag tree has a tag string tsn, which includes the tags of n and all tags of n's descendants, and a tag path tpn, which includes the tags from the root to n.

5.4 Data Region Identification Module

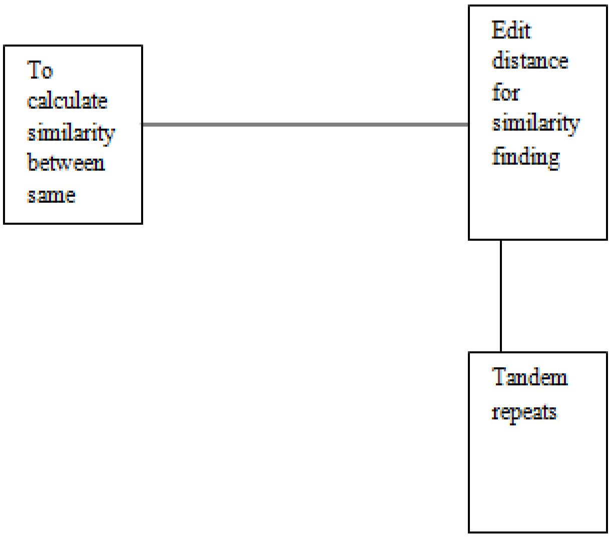
To calculate similarity between same
Edit distance for similarity finding
Tandem repeats

Figure 3: Data Region Identity

Some child sub trees of the same parent node form similar data records, which assemble a data region.

5.5 Record Segmentation Module

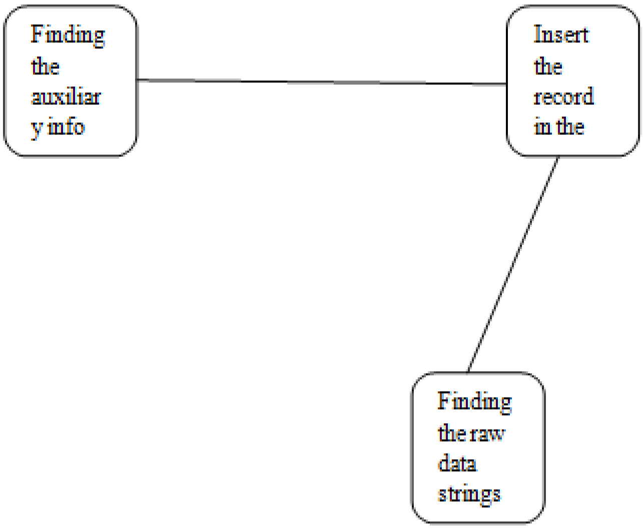

Figure 4: Segmenting Module

Record segmentation first finds tandem repeats within a data region. If only one tandem repeat is found in a data region, we assume that each repeated instance inside the tandem repeat corresponds to a record, if multiple tandem repeats are found in a data region we need to select one to denote the record.

5.6 Query Result Section Identification Module

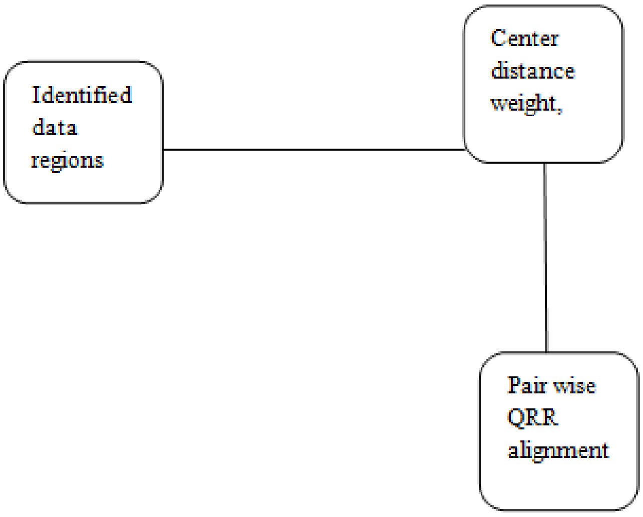

Figure 5: Identifying Data Region

Even after performing the data region merge step, there may still be multiple data regions in a query result page. However, we assume that at most one data region contains the actual QRRs. Three heuristics are used to identify this data region, called the query result section.

6. Conclusion And Future Enhancement

6.1 Conclusion

In this project a novel data extraction method, CTVS, to automatically extract QRRs from a query result page. CTVS employs two steps for this task. The first step identifies and segments the QRRs to improve the existing techniques by allowing the QRRs in a data region to be non contiguous. 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: pair wise alignment, holistic alignment, and nested structure processing. Experiments on five data sets show that CTVS is generally more accurate than current state-of-the-art methods.

6.2 Future Enhancement

CTVS has been shown to be an accurate data extraction method; it still suffers from some limitations. First, it requires at least two QRRs in the query result page. Second, any optional attribute that appears as the start node in a data region will be treated as auxiliary information. Third, similar to other related works, CTVS mainly depends on tag structures to discover data values. Therefore, CTVS does not handle the case where multiple data values from more than one attribute are clustered inside one leaf node of the tag tree, as well as the case where one data value of a single attribute spans multiple leaf nodes. Finally, as previously mentioned, if a query result page has more than one data region that contains result records and the records in the different data regions are not similar to each other, then CTVS will select only one of the data regions and discard the others.

References

[1] W.Cohen, M. Hurst, and L. Jensen, “A Flexible Learning System for Wrapping Tables and Lists in HTML Documents,” Proc. 11th World Wide Web Conf., pp. 232-241, 2002.

[2] L.Chen, H.M. Jamil, and N. Wang, “Automatic Composite Wrapper Generation for Semi-Structured Biological Data Based on Table Structure Identification,” SIGMOD Record, vol. 33, no. 2,pp. 58-64, 2004.

[3] A.Arasu and H. Garcia-Molina, “Extracting Structured Data from Web Pages,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 337-348, 2003.

[4] L.Chen, H.M. Jamil, and N. Wang, “Automatic Composite Wrapper Generation for Semi-Structured Biological Data Based on Table Structure Identification,” SIGMOD Record, vol. 33, no. 2,pp. 58-64, 2004.

[5] C.H.Chang and S.C. Lui, “IEPAD: Information Extraction Based on Pattern Discovery,” Proc. 10th World Wide Web Conf., pp. 681- 688, 2001.

[6] K.C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, “Structured Databases on the Web: Observations and Implications,” SIGMOD Record, vol. 33, no. 3, pp. 61-70, 2004.

[7] D. Buttler, L. Liu, and C. Pu, “A Fully Automated Object Extraction System for the World Wide Web,” Proc. 21st Int’l Conf. Distributed Computing Systems, pp. 361-370, 2001.

[8] R.Baeza-Yates, “Algorithms for String Matching: A Survey,” ACM SIGIR Forum, vol. 23, nos. 3/4, pp. 34-58, 1989.

[9] R.Baumgartner,S. Flesca, and G. Gottlob, “Visual Web Information Extraction with Lixto,” Proc. 27th Int’l Conf. Very Large Data Bases, pp. 119-128, 2001.

[10] M.K.Bergman,“TheDeepWeb: Surfacing Hidden Value,”White Paper,Bright Planet Corporation,.bright planet. com/ resources/details/deepweb.html, 2001.

Author Profile

Mr. M. Jude Victor received bachelor’s degree (Dr. M.G.R University) in Information Technology, Maduravoyal, Chennai, India in 2011 and doing master’s degree in Computer Science and Engineering in Hindustan University, Chennai.