Data Extraction and alignment for multiple web Databases

Anuradha R. Kale, Prof V.T.Gaikwaid, Prof H.N.Datir.

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. For this data extraction and alignment method are proposed. Data extraction from deep webs needs to be improved to achieve the efficiency and accuracy of automatic wrappers. For extraction CTVS that combines both tag and value similarity method are used to extract the data from multiple web databases. For Alignment re-ranking method are propose which employs semantic similarity to improve the quality of search results. Fetch the top N results returned by search engine, and use semantic similarities between the candidate and the query to re-rank the results. First convert the ranking position to an importance score for each candidate. Then combine the semantic similarity score with this initial importance score and finally get the new ranks.

Index Terms— Data extraction, data record alignment, information integration.

1 INTRODUCTION

Online databases, comprise the deep web. 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 semistructured, encoded in HTML pages.

Many web applications, such as metaquerying, 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.

The objective of this project is to extract data from multiple web data bases and align them in one format. Where anyone fire a query for they get a result from one particular database and it should be limited one. But if data come from multiple web databases, then it contain more results as compared to single database. The advantage of using multiple web database is that we get more relevant data .For this we used two databases Google and Faroo. WITH the advent of information technology, a user is able to obtain relevant information from the World Wide Web, which contains a huge amount of information, simply and quickly by entering search queries . In response to information and deliver it directly to the user.

2 Literature Survey & Related work:

Web database extraction has received much attention from the Database and Information Extraction research areas in recent years due to the volume and quality of deep web data. As the returned data for a query are embedded in HTML pages, the research has focused on how to extract this data. Earlier work focused on wrapper induction methods, which require human assistance to build a wrapper. More recently, data extraction methods have been proposed to automatically extract the records from the query result pages. In wrapper induction, extraction rules are derived based on inductive learning. A user labels or marks part or all of the item(s) to extract (the target item(s)) in a set of training pages or a list of data records in a page and the system then learns the wrapper rules from the labeled data and uses them to extract records from new pages. A rule usually contains two patterns, a prefix pattern and a suffix pattern, to denote the beginning and the end, respectively, of the target item. Some existing systems that employ wrapper induction include WIEN, SoftMealy, Stalker, XWRAP, WL2 and , and Lixto .

Related works on Web data extraction can be classified into three categories: 1) wrapper programming languages, 2) wrapper induction, and 3) automatic extraction. The first approach provides some specialized pattern specification languages to help the user construct extraction programs. Visual platforms are also provided to hide their complexities under simple graphical wizards and interactive processes. Systems that use this approach include WICCAP , Wargo, Lixto, DEBye, etc. The second approach is wrapper induction, which uses supervised learning to learn data extraction rules from a set of manually labeled examples. Manual labeling of data is labor intensive and time consuming. Furthermore, for different sites or even pages in the same site, the manual labeling process needs to be repeated because they may follow different templates. Example wrapper induction systems include WIEN , Softmealy, Stalker, WL2, Thresher, IDE , etc. Our technique requires no human labeling. The third approach is automatic extraction. Embley et al. proposes using a set of heuristics and domain ontologies to automatically identify data record boundaries. Buttler et al. proposes additional heuristics for the task without using domain ontologies.

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 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 using re-ranking method. This employs semantic similarity to improve the quality of search results. We fetch the top N results returned by search engine, and use semantic similarities between the candidate and the query to re-rank the results. We first convert the ranking position to an importance score for each candidate. Then we combine the semantic similarity score with this initial importance score and finally we get the new ranks.

5. Module Description

5.1 System Architecture

The general architecture of our system is given in Fig. 5.1. The input to the system is a Web page containing lists of data records (a page may contain multiple regions or areas with regularly structured data records). The system is composed of the following main components:

1. Google and Faroo Databases: From this Databases we extract the data for given input. Data from these databases GOOGLE API and Json API, are used, which returns the rendering information from respective databases.

2. Data Regions Identifier: Check the occurrence for input word identifies each area or region in the page that contains a list of similar data records.

3. Re-raking Method: After identifying the data region of similar record, using the importance score for each web page we find out the relevance of data.

4. Display result: After finding out the importance score, align the data in descending order from that score. This means most relevant data contain highest score and it will be display first.

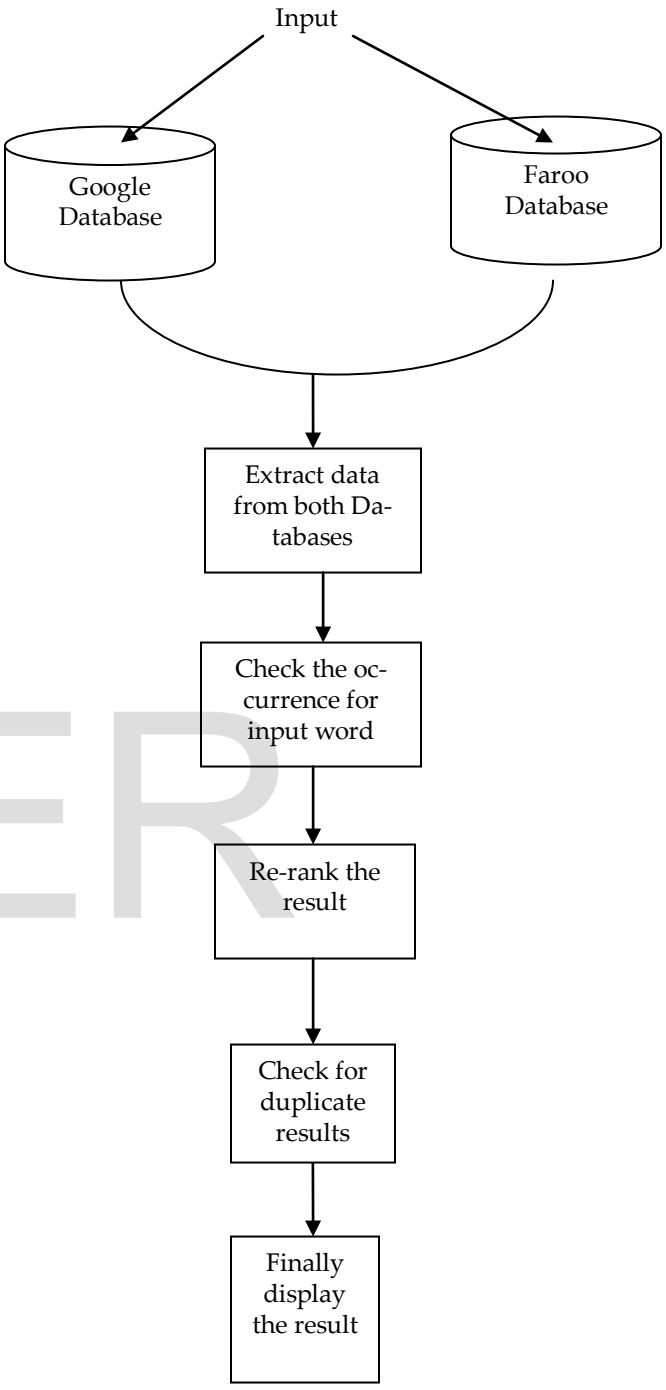

Figure 5.1 The general architecture of system

5.2 Google API

This tool is used to extract data from Google database. The Google API stands for 'Application Programmable Interface'. As its name implies, it is an interface that queries the Google database to help programmers in the development of their applications. Google API's consist basically of specialized Web services and programs and specialized scripts that

enable Internet application developers to better find and process information on the Web. In essence, Google APIs can be used as an added resource in their applications.

5.3 Json API

JSON has become a very popular lightweigth format for data exchange. JSON is human readable and easy for computers to parse and use. However, JSON is schemaless. Though this brings some benefits (e.g., flexibility in the representation of the data) it can become a problem when consuming and integrating data from different JSON services since developers need to be aware of the structure of the schemaless data. We believe that a mechanism to discover (and visualize) the implicit schema of the JSON data would largely facilitate the creation and usage of JSON services. For instance, this would help developers to understand the links between a set of services belonging to the same domain or API.

5.4 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.

5.5 Data Region Identification Module

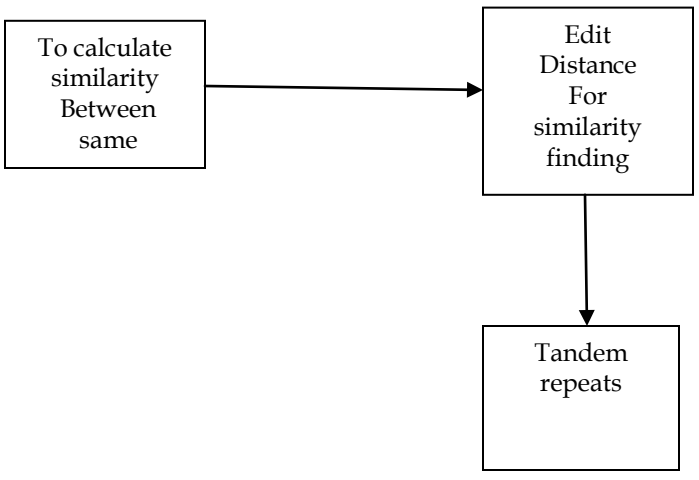

Figure 5.5: Data Region Identity

5.6 Record Segmentation Module

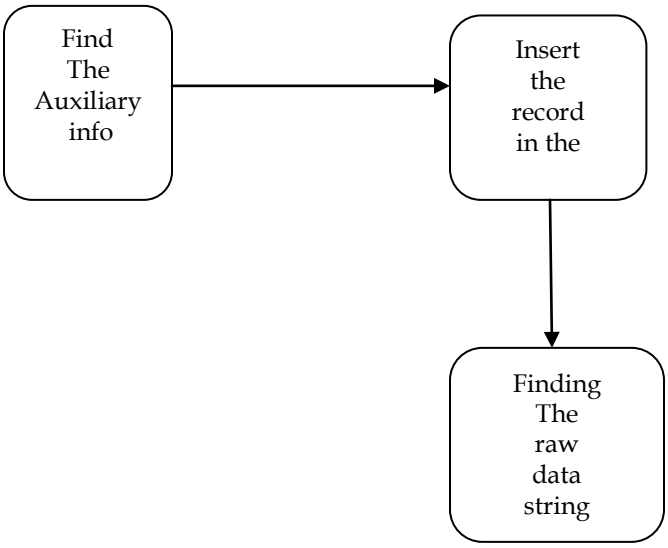

Figure 5.6: Segmenting Module

5.7 Re-ranking Module

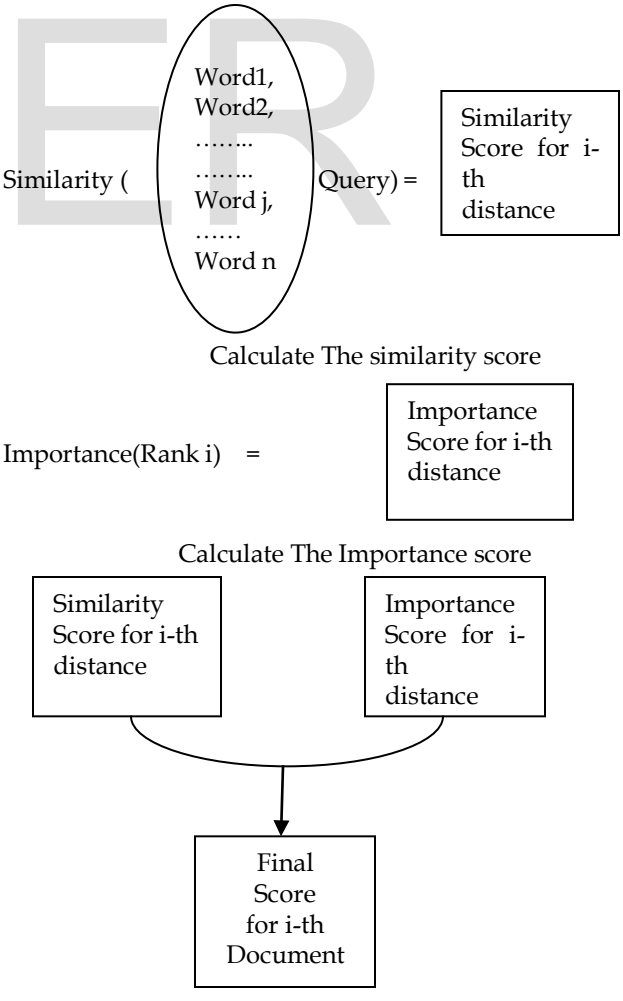

Figure 5.7: re-ranking method

6 Implemetion of Re-ranking Method

The semantic analysis method is use to remedy the shortcomings of the current search techniques. The search based on lexica semantics instead of keyword matching can better adapt to the thinking pattern of human beings, and thus search results are more relevant to users' search intention. Meanwhile, using semantic factors can conciliate the freshness and make the high-relevant new pages get moderate rank promotion. In our work, we fetch the top N results returned by search engines such as Google for user queries, and use semantic similarities between the candidate and the query to re-rank the results. First convert the ranking position to an importance score for each candidate. Then combine the semantic similarity score with this initial importance score and finally we get the new ranks. We analyze the combination ratio between these two parts and choose a best one. The experimental results validate that our proposed method can indeed improve the search performance.

6.1 Importance

since our result is heavily depended on the search engine's quality and result, how to grade the web pages returned from the search engine is important.

How to measure the importance of the results at different positions. As we know, the search results are returned by search engines according to their importance and relevance. The most important web pages usually are returned at the top positions, and hence attract much more attention from users. On the contrary, the unimportant pages are returned at the bottom positions. Therefore, a discount factor is needed which progressively reduces the document value as its rank decreases.

We propose the following formula to calculate each web's importance score.

$$importance\ (i) = \frac{1 - (i - 1) / tot}{log_2 (i + 1)}$$

where i is the original PageRank serial number (i.e.,original ranking position) and tot is the number of the fetched web pages for a query. The formula indicates that the top results have significant importance to the search keywords and thereby are much valuable for web users.

6.2 Step of Re-ranking Algorithm

1 Calculate the importance *(i)* for each web page which are extracted for result.
2 Arrange this rank of i in descending order
3 Now matched the title with USD, if matched then
Original rank $i + 1$;
4 If contain matched then
Original rank $i + 5$;
5 If url matched then
Original rank $i + 10$;
6 Finally we get result in descending order.

7. Conclusion

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. For this data extraction and alignment method are proposed. Data extraction from deep webs needs to be improved to achieve the efficiency and accuracy of automatic wrappers. For extraction CTVS that combines both tag and value similarity method is used to extract the data from multiple web databases. For Alignment re-ranking method is implemented which employs semantic similarity to improve the quality of search results. Fetch the top N results returned by search engine, and use semantic similarities between the candidate and the query to re-rank the results. First convert the ranking position to an importance score for each candidate. Then combine the semantic similarity score with this initial importance score and finally get the new ranks. This re-ranking method work on User profile Data (USD). After getting this new rank, we re-rank the data according to the relevance of USD

REFERENCES

[1] Weifeng Su, Jiying Wang, Frederick H. Lochovsky, "Combining Tag and Value Similarity for Data Extraction and Alignment" IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 7, July 2012

[2] Y. Zhai and B. Liu, "Structured Data Extraction from the WebBased on Partial Tree Alignment," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 12, pp. 1614-1628, Dec. 2006.

[3] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. 12th World Wide Web Conf. pp. 187-196, 2003.

[4] Ruofan Wang, Shan Jiang and Yan Zhang: Re-ranking Search Results Using Semantic Similarity .

[5] J. Balinski and C. Danilowicz. Re-ranking method based on inter-document distances. Information Processing and Management, 41(2005), pages 759–775, 2005.

[6] Nambiar, U., and Kambhampati, S. Providing Ranked Relevant Results for Web Database Queries. In Proceedings of the World Wide Web Conference, pp. 314-315. 2004.

[7] W. Su, J. Wang, and F.H. Lochovsky, "Holistic Schema Matching for Web Query Interfaces," Proc. 10th Int'l. Conf. Extending Database Technology, pp. 77-94, 2006

[8] C. Tao and D.W. Embley, "Automatic Hidden-Web Table Interpretation by Sibling Page Comparison," Proc. 26th Int'l Conf. Conceptual Modeling, pp. 566-581, 2007

[9] R. Baxter, P. Christen, and T. Churches, "A Comparison of Fast Blocking Methods for Record Linkage," Proc. KDD Workshop Data Cleaning, Record Linkage, and Object Consolidation, pp. 25-27, 2003.

[10] K. Simon and G. Lausen, "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions," Proc. 14th ACM Int'l Conf. Information and Knowledge Management, pp. 381-388, 2005.

[11] 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.

[12] J. Wang and F. Lochovsky, "Data-Rich Section Extraction from HTML Pages," Proc. Third Int'l Conf. Web Information System Eng., 2002

[13] 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.

[14] Yu, J. Han, and C.C. Chang, "PEBL: Web Page Classification without Negative Examples," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 1, pp. 70-81, Jan. 2004.

[15] 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.

[16] P. Bonizzoni and G.D. Vedova, "The Complexity of Multiple Sequence Alignment with SP-Score that Is a Metric," Theoretical Computer Science, vol. 259, nos. 1/2, pp. 63-79, 2001.

[17] Weifeng Su, Jiying Wang, Frederick H. Lochovsky, "Record Matching over Query Results from Multiple Web Databases" IEEE Transactions On Knowledge And Data Engineering, Vol. 22, No. 4, April 2010