Multiple Web Database Handle Using CTVS Method and Record Matching

Harish Chaware[#1],Prof. Nitin Chopade [#2]
[#1]M.E (Scholar),
G.H.Raisoni College of Engineering and Management, Amravati.
[*2]M.E (CSE),
G.H.Raisoni College of Engineering and Management, Amravati.

Abstract

Web databases generate query result pages based on a user's query. For many applications, automatically extracting the data from these query result pages is very important, such as data integration, which needs 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 present an unsupervised, online record matching method, UDD, which, for a given query, can effectively identify duplicates from the query result records of multiple Web databases. We propose new techniques to handle the case when the QRRs are not contiguous, which may be due to the presence of auxiliary information, such as a comment, recommendation or advertisement, and for handling any nested structure that may exist in the QRRs.

Keyword: Data extraction, automatic wrapper generation, data record alignment, information integration, Record matching, duplicate detection, record linkage.

I. Introduction

Today, more and more databases that dynamically generate Web pages are available on the Web in response to user queries. These Web databases compose the deep or hidden 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 semi structured, encoded in HTML pages.[1] Compare the query results returned from multiple Web databases, a crucial task is to match the different sources' records that refer to the same real-world entity. The records to match are highly query-dependent, since they can only be obtained through online queries. Moreover, they are only a partial and biased portion of all the data in the source Web databases [2].

This seminar focuses on the problem of identifying duplicates, that is, two records describing the same entity, has attracted much attention from many research fields, including Databases, Data Mining, Artificial Intelligence, and Natural Language Processing, and 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 values belonging to the same attribute are placed into the same table column. We propose two new techniques: Record matching and CTVS.[3]

Unsupervised Duplicate Detection (UDD)

Record matching method Unsupervised Duplicate Detection (UDD) for the specific record matching problem of identifying duplicates among records in query results from multiple Web databases. The key ideas of our method are:

1. We focus on techniques for adjusting the weights of the record fields in calculating the similarity between two records. Two records are considered as duplicates if they are "similar enough" on their fields. We believe different fields may need to be assigned different importance weights in an adaptive and dynamic manner.
2. We use a sample of universal data consisting of record pairs from different data sources as an approximation for a negative training set as well as the record pairs from the same data source. [4]

Combining Tag and Value Similarity (CTVS)

We employ the following two-step method, called Combining Tag and Value

Similarity (CTVS), to extract the QRRs from a query result page p.

- Record extraction identifies the QRRs in p and involves two substeps: data region identification and the actual segmentation step.
- Record alignment aligns the data values of the QRRs in p into a table so that data values for the same attribute are aligned into the same table column.

- *CTVS improves data extraction accuracy in three ways:*

1. New techniques are proposed to handle the case when the QRRs are not contiguous in p, which may be due to the presence of an auxiliary information, such as a comment, recommendation, or advertisement. While the method in can find all data regions containing at least two QRRs in a query result page using data mining techniques, almost all other data extraction methods, such as, assume that the QRRs are presented contiguously in only one data region in a page. However, this assumption may not be true for many web databases where auxiliary information separates the QRRs.[5] We examined 100 websites to determine the extent to which the QRRs in the query result pages are noncontiguous. We found that the QRRs in 26 out of the 100 websites are noncontiguous, which indicates that noncontiguous data regions are quite common. Furthermore, 14 of the 26 websites have noncontiguous QRRS with the same parent in the page HTML tag tree. The other 12 websites have noncontiguous QRRs with different parents in the HTML tag tree. To address this problem, we employ two techniques according to the layout of the QRRs and the auxiliary information in the result page's HTML tag trees.

a. An adapted data region identification method is proposed to identify the noncontiguous QRRs that have the same parents according to their tag similarities.

b. A merge method is proposed to combine different data regions that contain the QRRs (with or without the same parent) into a single data region.

2. A novel method is proposed to align the data values in the identified QRRs, first pairwise then holistically, so that they can be put into a table with the data values belonging to the same attribute arranged into the same table column. Both tag structure similarity and data value similarity are used in the pairwise alignment process. To our knowledge, this work is the first to combine tag structure and data value similarity to perform the alignment. We observe that the data values within the same attribute usually have the same data type, and similar data values in many cases, because they are the result for the same query. Hence, the pairwise alignment is reduced to finding a value-to-value alignment with the maximal data value similarity score under some constraints. After all pairs of records are aligned, a holistic alignment is performed, by viewing the pairwise alignment result as a graph and finding the connected components from the graph.

3. A new nested-structure processing algorithm is proposed to handle any nested structure in the QRRs after the holistic alignment. Unlike existing nested-structure processing algorithms that rely on only tag information, CTVS uses both tag and data value similarity information to improve nestedstructure processing accuracy.

II. Related work:

An important aspect of duplicate detection is to reduce the number of record pair comparisons. Several methods have been proposed for this purpose including standard blocking [1], sorted neighborhood method [2], Bigram Indexing, and record clustering. Even though these methods differ in how to partition the data set into blocks, they all considerably reduce the number of comparisons by only comparing records from the same block. Since any of these methods can be incorporated into UDD to reduce the number of record pair comparisons, we do not further consider this issue. While most previous record matching work is targeted at matching a single type of record, more recent work[1], [3],[5] has addressed the matching of multiple types of records with rich associations between the records. Even though the matching complexity increases rapidly with the number of record types, these works manage to capture the matching dependencies between multiple record types and utilize such dependencies to improve the matching accuracy of each single record type. Unfortunately, however, the dependencies among multiple record types are not available for many domains. Compared to these previous works, UDD is specifically designed for the Web database scenario where the records to match are of a single type with multiple string fields. These records are heavily query-dependent and are only a partial and biased portion of the entire data, which makes the existing work based on offline learning inappropriate. Moreover, our work focuses on studying and addressing the field weight assignment issue rather than on the similarity measure.

In UDD, any similarity measure, or some combination of them, can be easily incorporated. Our work is also related to the classification problem using only a single class of training examples, i.e., either positive or negative, to find data similar to the given class. To date, most single-class classification work has relied on learning from positive and unlabeled data [4],[5],[7]. In [9], multiple classification methods are compared and it is concluded that one-class SVM and neural network methods are comparable and superior to all the other methods. In particular, one-class SVM distinguishes one class of data from another by drawing the class boundary of the provided data's class in the feature space. However, it requires lots of data to induce the boundary precisely, which makes it liable to overfit or underfit the data and, moreover, it is very vulnerable to noise. The record matching works most closely related to UDD are Christen's method [10] and PEBL [11]. Using a nearest based approach, Christen first performs a comparison step to generate weight vectors for each pair of records and selects those weight vectors as training examples that, with high likelihood, correspond to either true matches (i.e., pairs with high similarity scores that are used as positive examples) or true nonmatches (i.e., pairs with low similarity scores that are used as negative examples). These training examples are then used in a convergence step to train a classifier (either nearest neighbor or SVM) to label the record pairs not in the training set. Combined, these two steps allow fully automated, unsupervised record pair classification, without the need to know the true match and nonmatch status of the weight vectors produced in the comparison step. PEBL [14] classifies Web pages in two stages by learning from positive examples and unlabeled data. In the mapping stage, a weak classifier, e.g., a rule-based one, is used to get "strong" negative examples from the unlabeled data, which contain none of the frequent features of the positive examples. In the convergence stage, an internal classifier, e.g., SVM, is first trained by the positive examples and the autoidentified negative examples and is then used to iteratively identify new negative examples until it converges.

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 [12], [13], [14], and [16]. 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 [1], Stalker [2] and [3], XWRAP [4], WL2 [7] and [10], and Lixto [13]. While wrapper induction has the advantage that no extraneous data are extracted, since the user can label only the items of interest, it requires labor intensive and time-consuming manual labeling of data. Thus, it is not scalable to a large number of web databases. Moreover, an existing wrapper can perform poorly when the format of a query result page changes, which may happen frequently on the web. Hence, the wrapper induction approach involves two further difficult problems: monitoring changes in a page's format and maintaining a wrapper when a page's format changes. To overcome the problems of wrapper induction, some unsupervised learning methods, such as RoadRunner, Omini, IEPAD, ExAlg [1], DeLa [2], PickUp [9], and TISP [10], have been proposed to automatically extract the data from the query result pages. These methods rely entirely on the tag structure in the query result pages. Here, we discuss only DeLa since we compare its performance with CTVS. DeLa models the structured data contained in template-generated webpages as string instances encoded in HTML tags, of the implied nested type of their web database. A regular expression is employed to model the HTML-encoded version of the nested type. Since the HTML tag-structure enclosing the data may appear repeatedly if the page contains more than one instance of the data, the page is first transformed into a token sequence composed of HTML tags and a special token "text" representing any text string enclosed by pairs of HTML tags. Then, continuous repeated substrings are extracted from the token sequence and a regular expression wrapper is induced from the repeated substrings according to some hierarchical relationships among them. The main problem with this method is that it often produces multiple patterns (rules) and it is hard to decide which is correct.

III. Proposed Work:

In this section, we describe the methods that are use in Data extraction and Alignment. There is method for automatically extracting data records that are encoded in the query result pages generated by web databases. [3]

1 QRR Extraction

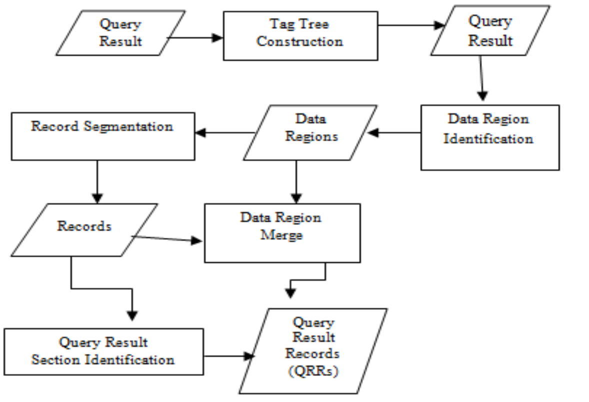

Figure 1. QRR extraction framework.

Fig. 1 shows the framework for QRR extraction. Given a query result page, the 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. Next, the Data Region Identification module identifies all possible data regions, which usually contain dynamically generated data, top down starting from the root node. The Record Segmentation module then segments the identified data regions into data records according to the tag patterns in the data regions. Given the segmented data records, the Data Region Merge module merges the data regions containing similar records. Finally, the Query Result Section Identification module selects one of the merged data regions as the one that contains the QRRs. The following four sections describe each of the last four modules in detail.[6]

1. Data Region Identification

We first assume that some child subtrees of the same parent node form similar data records, which assemble a data region. which claim that similar data records are typically presented contiguously in a page. Instead, we observed that in many query result pages some additional item that explains the data records, such as a recommendation or comment, often separates similar data records. Hence, we propose a new method to handle non-contiguous data regions so that it can be applied to more web databases.

2. Record Segmentation

To illustrate the record segmentation algorithm, first finds tandem repeats within a data region. The following two heuristics are used for the tandem repeat selection:

- If there is auxiliary information, which corresponds to nodes between record instances, within a data region, the tandem repeat that stops at the auxiliary information is the correct tandem repeat since auxiliary information usually is not inserted into the middle of a record.
- The visual gap between two records in a data region is usually larger than any visual gap within a record. Hence, the tandem repeat that satisfies this constraint is selected.
- If the preceding two heuristics cannot be used, we select the tandem repeat that starts the data region.[7]

3. Data Region Merge

The data region identification step may identify several data regions in a query result page. Moreover, the actual data records may span several data regions. In the websites we examined, 12 percent had QRRs with different parents in the HTML tag tree. Thus, before we can identify all the QRRs in a query result page, we need to determine whether any of the data regions should be merged. Given any two data regions, we treat them as similar if the segmented records they contain are similar. The similarity between any two records from two data regions is measured by the similarity of their tag strings. The similarity between two data regions is calculated as the average record similarity.

4. Query Result Section Identification

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.

- The query result section usually occupies a large space in the query result page
- The query result section is usually located at the center of the query result page
- Each QRR usually contains more raw data strings than the raw data strings in other sections.

4.1 QRR Alignment

QRR alignment is performed by a novel three-step:

1. Pair wise QRR Alignment

The pair wise QRR alignment algorithm is based on the observation that the data values belonging to the same attribute usually have the same data type and may contain similar strings, especially since the QRRs are for the same query. During the pairwise alignment, we require that the data value alignments must satisfy the following three constraints:

- Same record path constraint

- unique constraint
- No cross alignment constraint

2. Holistic Alignment

Given the pair wise data value alignments between every pair of QRRs, the step of holistic alignment performs the alignment globally among all QRRs to construct a table in which all data values of the same attribute are aligned in the same table column. Intuitively, if we view each data value in the QRRs as a vertex and each pairwise alignment between two data values as an edge, the pairwise alignment set can be viewed as an undirected graph.

3. Nested Structure Processing

Holistic data value alignment constrains a data value in a QRR to be aligned to at most one data value from another QRR. If a QRR contains a nested structure such that an attribute has multiple values, then some of the values may not be aligned to any other values. Therefore, nested structure processing identifies the data values of a QRR that are generated by nested structures. To overcome this problem, CTVS uses both the HTML tags and the data values to identify the nested structures.

4.3 UDD

Our focus is on Web databases from the same domain, i.e., Web databases that provide the same type of records in response to user queries. Suppose there are s records in data source A and there are t records in data source B, with each record having a set of fields/attributes. Each of the t records in data source B can potentially be a duplicate of each of the s records in data source A. The goal of duplicate detection is to determine the matching status, i.e., duplicate or nonduplicate, of these s * t record pairs.

We present the assumptions and observations on which UDD is based.

1. A global schema for the specific type of result records is predefined and each database's individual query result schema has been matched to the global schema.
2. Record extractors, i.e., wrappers, are available for each source to extract the result data from HTML pages and insert them into a relational database according to the global schema.

Besides these two assumptions, we also make use of the following two observations:

1. The records from the same data source usually have the same format.
2. Most duplicates from the same data source can be identified and removed using an exact matching method.

Duplicate records exist in the query results of many Web databases, especially when the duplicates are defined based on only some of the fields in a record. Using a straightforward pre-processing step, exact matching, can merge those records that are exactly the same in all relevant matching fields. [10]

IV. Conclusion

We presented 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 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: pairwise 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.

Duplicate detection is an important step in data integration and most state-of-the-art methods are based on offline learning techniques, which require training data. In the

Web database scenario, where records to match are greatly query-dependent, a pretrained approach is not applicable as the set of records in each query's results is a biased subset of the full data set. To overcome this problem, we presented an unsupervised, online approach, UDD, for detecting duplicates over the query results of multiple Web databases. Two classifiers, WCSS and SVM, are used cooperatively in the convergence step of record matching to identify the duplicate pairs from all potential duplicate pairs iteratively. Experimental results show that our approach is comparable to previous work that requires training examples for identifying duplicates from the query results of multiple Web databases.

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] 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

[3] 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.

[4] 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

[5] 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

[6] 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.

[7] 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.

[8] 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.

[9] J. Wang and F. Lochovsky, “Data-Rich Section Extraction from HTML Pages,” Proc. Third Int’l Conf. Web Information System Eng., 2002

[10] 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.