Data Extraction from Web Database Search Result Using Automatic Annotation

B.Srujana *, G.UdayKiran
Mtech Student, CSE Dept, BVRIT, srujana1253@gmail.com
AssistantProfessor, CSE Dept, BVRIT udaykiran.goru@bvrit.ac.in

Abstract: - Data extraction has become the need of the hour with so much data being populated on web pages. More specifically, the users expect the data extraction to be structured and dynamic. Data extraction from the HTML based search interfaces is usually performed by wrappers. Manually wrapping results into poor scalability. The querying process through the search interfaces retrieves data from multiple data units corresponds to one semantic and the result set is referred to as search result records (SRRs). Now data has to be extracted from these SRRs and assign meaningful labels referred as Annotation. The annotated data units with same semantic meaning are grouped. An annotation wrapper can be used to annotate the new result records from the same web database. In this paper we present the automatic annotation approach which involve three phases to annotate and display the result.

Keywords: Result record, web database, data units, annotation wrapper

1 Introduction

The Web has become the preferred medium for many database applications, such as e-commerce and digital libraries. These applications store information in huge databases that user’s access, query, and update through the Web. Database-driven Web sites have their own interfaces and access forms for creating HTML pages on the fly. Web database technologies define the way that these forms can connect to and retrieve data from database servers. The number of database-driven Websites is increasing exponentially, and each site is creating pages dynamically—pages that are hard for traditional search engines to reach. Such search engines crawl and index static HTML pages; they do not send queries to Web databases.

The encoded data units to be machine process able, which is essential for many applications such as deep web data collection and Internet comparison shopping, they need to be extracted out and assigned meaningful labels.

The explosive growth and popularity of the World Wide Web has resulted in a huge amount of information sources on the Internet. However, due to the heterogeneity and the lack of structure of Web information sources, access to this huge collection of information has been limited to browsing and searching. Sophisticated Web mining applications, such as comparison shopping robots, require expensive maintenance to deal with different data formats. To automate the translation of input pages into structured data, a lot of efforts have been devoted in the area of information extraction (IE). A typical result page returned from a WDB has multiple search result records (SRRs). Each SRR contains multiple data unit search of which describes one aspect of a real-world entity. In this paper, a 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 paper, 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 different book sites, it needs to determine whether any two SRRs refer to the same book. We propose a clustering-based shifting technique to align data units into different groups so that the data units inside the same group have the same semantic. Instead of using only the DOM tree or other HTML tag tree structures of the SRRs to align the data units (like most current methods do), our approach also considers other important features shared among data units, such as their data types (DT), data contents (DC), presentation styles (PS), and adjacency (AD) information.

2. Implementation

Our automatic annotation solution consists of three phases as

![Fig 1: Phases of automatic annotation solution](image)

1) Extracts (automatically) text from a webpage into a table
2) Assigns labels in a table.

**Phase 1** is the alignment phase. In this phase, we first identify all data units in the search records and then organize them into different groups with each group corresponding to a different concept. The result of this phase with each column containing data units of the same concept across all search records. Grouping data units of the same meaning can help identify the common patterns and features among these data units. These common features are the basis of our annotators.

**Phase 2** is the annotation phase. We introduce multiple basic annotators with each exploiting one type of features. Every basic annotator is used to produce a label for the units within their group holistically, and a probability model is adopted to determine the most appropriate label for each group.

**Phase 3** is the annotation wrapper generation. In this phase, we generate an annotation rule that describes how to extract the data units of this concept in the result page and what the appropriate meaning annotation should be. The rules for all aligned groups, collectively, form the annotation wrapper for the corresponding WDB, which can be used to directly assign label the data retrieved from the same WDB in response to new queries without the need to perform the above two phases again. As such, annotation wrappers can perform annotation quickly, which is essential for online applications.

3. Alignment Phase

3.1. Features of data units

Data units belonging to the same concept from different SRRs usually share many common features. Five features are utilized in our approach.

1. **Data Content (DC).** The data units with the same concept often share certain keywords. This is true for several reasons. First, the data units corresponding to the search field where the user enters a search condition usually contain the search keywords.

2. **Presentation Style (PS).** This feature describes how a data unit is displayed on a web page. It consists of 6 style features:
font face, font size, font color, font weight, text decoration (underline, strike, etc.), and whether it is italic. Data units of the same concept in different SRRs are usually displayed in the same style.

3. Data Type (DT). Each data unit has its own semantic type although it is just a text string in the HTML code. Seven basic data types are considered in our approach: Date, Time, Currency, Integer, Decimal, Percentage, and Ordinary String. Each type except Ordinary String has certain pattern(s) so that it can be easily identified. Text not one of the first 6 types is treated as an Ordinary String.

4. Tag Path (TP). A tag path of a data unit is a sequence of tags traversing from the root of the record to the corresponding node in the tag tree.

5. Adjacency (AD). Consider two data units d1 and d2 from two different SRRs r1 and r2, respectively. Let pi and si be the data units that precede and succeed di in ri, respectively, i = 1, 2. It can be observed that if p1 and p2 belong to the same concept and/or s1 and s2 belong to the same concept, then it is more likely that d1 and d2 also belong to the same concept.

3.2. Text nodes and data units
The visible elements on the web page represent a text node and the data units are located in the text nodes. Relationships between text node and data unit features are,
1. One-to-One Relationship: Text node containing exactly 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 tags <A> and </A>. This type of text nodes are referred as atomic text nodes. An atomic text node is equivalent to a data unit.
2. One-to-Many Relationship: Multiple data units are encoded in one text node. This type of text nodes are referred as composite text node.
3. Many-to-One Relationship: Multiple text nodes together form a data unit. This type of text nodes is referred as decorative tags because they are used for changing the appearance of part of the text node.
4. One-To-Nothing Relationship: Text nodes are not part of any data unit inside SRRs. This type of text node is referred as template text node.

4. Alignment Algorithm
Our data alignment algorithm is based on the assumption that attributes appear in the same order across all SRRs on the same result page, although the SRRs may contain different sets of attributes (due to missing values). The goal of alignment is to move the data units in the table so that every alignment group is well aligned, while the order of the data units within every SRR is preserved. Our data alignment method consists of the following four steps. The detail of each step will be provided later.

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.

The same algorithm is used in the first and the third steps above, with the only difference being that for the former text
nodes are considered while for the latter data units are considered. More specifically, the similarity between data units d1 and d2 is:

\[
\text{Sim}(d_1, d_2) = w_1*\text{SimC}(d_1, d_2) + w_2*\text{SimP}(d_1, d_2) + w_3*\text{SimD}(d_1, d_2) + w_4*\text{SimT}(d_1, d_2) + w_5*\text{SimA}(d_1, d_2).
\]

The feature similarities are defined as follows:

**Data content similarity (SimC):** It is the Cosine similarity [26] between texts of the two data units.

**Presentation style similarity (SimP):** It is the ratio of the number of style features the two data units match over the six style features used in our approach.

**Data type similarity (SimD):** If two data units have the same data type, the similarity is 1; otherwise, 0.

**Tag path similarity (SimT):** This is the edit distance between the tag paths of d1 and d2. The edit distance (EDT) refers to the number of insertions and deletions needed to transform one tag path into the other. Let t1 and t2 be the tag paths of d1 and d2 respectively,

\[
\text{SimT}(d_1, d_2) = 1 - \frac{\text{EDT}(t_1, t_2)}{\text{Len}(t_1) + \text{Len}(t_2)},
\]

where \(\text{Len}(t)\) denotes the number of tags in tag path \(t\).

**Adjacency similarity (SimA):** This is the average similarity between the preceding data units and between the succeeding units of d1 and d2.

We apply the agglomerative clustering algorithm to cluster the data units. Initially, each data unit forms a separate group of its own. We then repeatedly merge two groups that have the highest similarity value until no two groups have similarity above a threshold \(T\). After the splitting is completed for a composite group, the data units in this group are not aligned yet. Using these separators to generate the alignment directly may be problematic because the “values” in the composite text nodes often do not have a uniform format, for example, some data units may be missing in some of these “values” if the corresponding SRRs do not have information for some attributes. Our solution to the data unit alignment problem is to apply the above agglomerative clustering method to the data units in each composite group.

5. Assigning Labels

5.1 Local vs. integrated interface schemas

For a Web database, its local search interface often contains some attributes of the underlying data. We denote a local search interface schema (LIS) as \(\text{Si} = \{A_1, A_2, \ldots, A_k\}\), 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 \(\text{Se} = \{a_1, a_2, \ldots, a_n\}\), where each \(a_j\) (\(j = 1, \ldots, n\)) is an attribute to be discovered. The schema of the retrieved data and the interface schema usually share a significant number of attributes. We use WISE-Integrator as the basic tool to automatically build an integrated interface schema (IIS) over multiple Web databases in that domain. The generated integrated interface combines all the attributes of the local interface schemas. For each Web database in a given domain, our approach uses both the LIS of the database and the IIS of the domain to annotate the retrieved data units. Using IIS has two major advantages. First, it has the potential to increase the annotation recall. Since the integrated interface contains the attributes on 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 on the local interface. Second, when an annotator discovers a label for a group of data units, the label will be replaced with its corresponding global attribute name (if any) on the IIS by looking up the attribute-mapping table so that the data units of
the same concept across different Web databases will have the same label.

5.2. Basic annotators

The data units belonging to the same concept (attribute) often share special common features, which are usually displayed in certain patterns. Based on this observation, we define 6 basic annotators to label data units, with each of them considering a special type of patterns/features.

**Table Annotator (TA):** Many Web databases use a table to organize the returned SRRs, which visually has multiple rows and columns with each row representing an SRR. The table header, which indicates the meaning of each column, is usually located at the top of the table.

**Query-based Annotator (QA):** The basic idea of this annotator is that the returned SRRs from a Web database are always related to the specified query. Specifically, the query terms entered through the search attributes on the local search interface of the Web database will most likely appear in some retrieved SRRs. We use a similar method to evaluate the significance of each annotator. Each time, one annotator is removed and the remaining annotators are used to annotate the pages in Data Search.

6. Conclusion

Assigning meaningful labels to the extracted data unit of each SRR is a challenging task. The automatic annotation approach considers several types of data unit and text node features and makes annotation scalable and automatic.

Multiple annotators of different features are used to annotate the extracted information from the result pages. Each annotators exhibit one special type of feature and they are together used to automatically construct a high quality annotation wrapper. Uses both LIS and IIS for label assignment and alleviates local interface schema inadequacy and inconsistent label problem.

7. References


