JP INFOTECH
SOFTWARE DEVELOPMENT & RESEARCH DIVISION
www.jpinfotech.org
(0)9952649690

Chennai Office : JP INFOTECH, Old No.31, New No. 86, 1st Floor , 1st Avenue , Ashok Pillar , Chennai - 83.
Landmark : Next to Kotak Mahendra Bank.

Pondicherry Office : JP INFOTECH , #45 , Kamaraj Salai, Thattanchavady, Puducherry
Landmark : Next to VVP Nagar Arch.
jpinfotechprojects@gmail.com

Annotating Search Results from Web Databases

ABSTRACT:

An increasing number of databases have become web accessible through HTML form-based search interfaces. The data units returned from the underlying database are usually encoded into the result pages dynamically for human browsing. For 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. In this paper, 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.

EXISTING SYSTEM:

In this existing system, 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. It describes the relationships between text nodes and data units in detail. 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.

DISADVANTAGES OF EXISTING SYSTEM:

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. For instance, no 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.

PROPOSED SYSTEM:

In this paper, we consider how to automatically assign labels to the data units within the SRRs returned from WDBs. Given a set of SRRs that have been

extracted from a result page returned from a WDB, our automatic annotation solution consists of three phases.

ADVANTAGES OF PROPOSED SYSTEM:

This paper has the following contributions:

- While most existing approaches simply assign labels to each HTML text node, we thoroughly analyze the relationships between text nodes and data units. We perform data unit level annotation.
- 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.
- We utilize the integrated interface schema (IIS) over multiple WDBs in the same domain to enhance data unit annotation. To the best of our knowledge, we are the first to utilize IIS for annotating SRRs.
- We employ six basic annotators; each annotator can independently assign labels to data units based on certain features of the data units. We also

employ a probabilistic model to combine the results from different annotators into a single label. This model is highly flexible so that the existing basic annotators may be modified and new annotators may be added easily without affecting the operation of other annotators.

- We construct an annotation wrapper for any given WDB. The wrapper can be applied to efficiently annotating the SRRs retrieved from the same WDB with new queries.

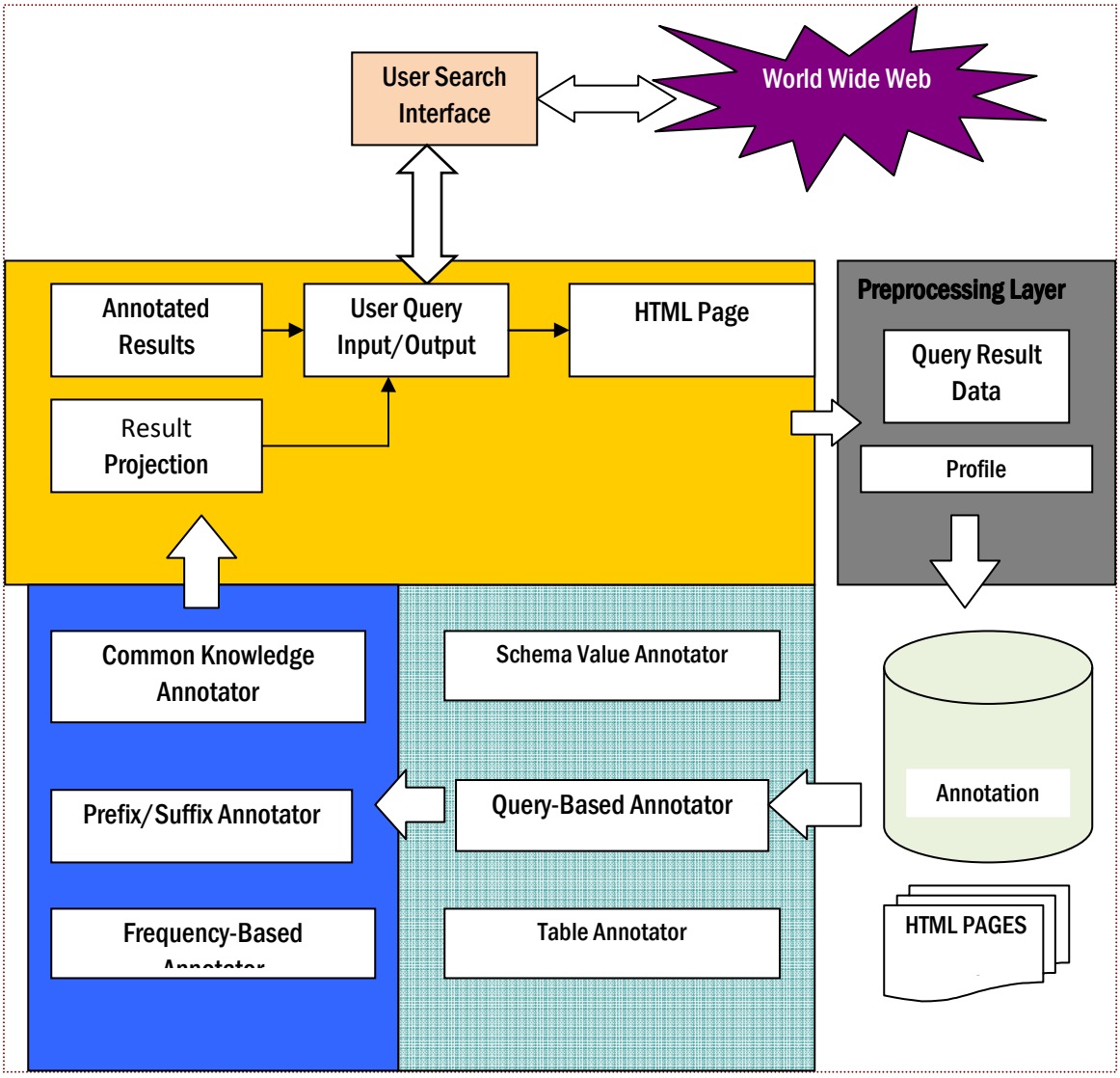
JP INFOTECH
SOFTWARE DEVELOPMENT & RESEARCH DIVISION
www.jpinfotech.org
(0)9952649690

Chennai Office : JP INFOTECH, Old No.31, New No. 86, 1st Floor , 1st Avenue , Ashok Pillar , Chennai - 83.
Landmark : Next to Kotak Mahendra Bank.

Pondicherry Office : JP INFOTECH , #45 , Kamaraj Salai, Thattanchavady, Puducherry
Landmark : Next to VVP Nagar Arch.

jpinfotechprojects@gmail.com

PROPOSED SYSTEM ARCHITECTURE:



JP INFOTECH
SOFTWARE DEVELOPMENT & RESEARCH DIVISION
www.jpinfotech.org
(0)9952649690

Chennai Office : JP INFOTECH, Old No.31, New No. 86, 1st Floor , 1st Avenue , Ashok Pillar , Chennai - 83.
Landmark : Next to Kotak Mahendra Bank.

Pondicherry Office : JP INFOTECH , #45 , Kamaraj Salai, Thattanchavady, Puducherry
Landmark : Next to VVP Nagar Arch.

jpinfotechprojects@gmail.com

MODULES:

- Basic Annotators
- Query-Based Annotator
- Schema Value Annotator
- Common Knowledge Annotator
- Combining Annotators

MODULES DESCRIPTION:

Basic Annotators

In a returned result page containing multiple SRRs, the data units corresponding to the same concept (attribute) often share special common features. And such common features are usually associated with the data units on the result page in certain patterns. Based on this observation, we define six basic annotators to label data units, with each of them considering a special type of patterns/features. Four of these annotators (i.e., table annotator, query-based annotator, intext prefix/suffix annotator, and common knowledge annotator) are similar to the annotation heuristics

Query-Based Annotator

The basic idea of this annotator is that the returned SRRs from aWDBare always related to the specified query. Specifically, the query terms entered in the search attributes on the local search interface of the WDB will most likely appear in some retrieved SRRs. For example, query term “machine” is submitted through the Title field on the search interface of the WDB and all three titles of the returned SRRs contain this query term. Thus, we can use the name of search field Title to annotate the title values of these SRRs. In general, query terms against an attribute may be entered to a textbox or chosen from a selection list on the local search interface. Our Query-based Annotator works as follows: Given a query with a set of query terms submitted against an attribute A on the local search interface, first find the group that has the largest total occurrences of these query terms and then assign gn(A) as the label to the group.

Schema Value Annotator

Many attributes on a search interface have predefined values on the interface. For example, the attribute Publishers may have a set of predefined values (i.e., publishers) in its selection list. More attributes in the IIS tend to have predefined values and these attributes are likely to have more such values than those in LISs, because when attributes from multiple interfaces are integrated, their values are

also combined. Our schema value annotator utilizes the combined value set to perform annotation.

The schema value annotator first identifies the attribute Aj that has the highest matching score among all attributes and then uses gn(Aj) to annotate the group Gi. Note that multiplying the above sum by the number of nonzero similarities is to give preference to attributes that have more matches (i.e., having nonzero similarities) over those that have fewer matches. This is found to be very effective in improving the retrieval effectiveness of combination systems in information retrieval

Common Knowledge Annotator

Some data units on the result page are self-explanatory because of the common knowledge shared by human beings. For example, “in stock” and “out of stock” occur in many SRRs from e-commerce sites. Human users understand that it is about the availability of the product because this is common knowledge. So our common knowledge annotator tries to exploit this situation by using some predefined common concepts.

Each common concept contains a label and a set of patterns or values. For example, a country concept has a label “country” and a set of values such as

"U.S.A.," "Canada," and so on. It should be pointed out that our common concepts are different from the ontologies that are widely used in some works in Semantic Web. First, our common concepts are domain independent. Second, they can be obtained from existing information resources with little additional human effort.

Combining Annotators

Our analysis indicates that no single annotator is capable of fully labeling all the data units on different result pages. The applicability of an annotator is the percentage of the attributes to which the annotator can be applied. For example, if out of 10 attributes, four appear in tables, then the applicability of the table annotator is 40 percent. The average applicability of each basic annotator across all testing domains in our data set. This indicates that the results of different basic annotators should be combined in order to annotate a higher percentage of data units. Moreover, different annotators may produce different labels for a given group of data units. Therefore, we need a method to select the most suitable one for the group. Our annotators are fairly independent from each other since each exploits an independent feature.

SYSTEM CONFIGURATION:-

HARDWARE CONFIGURATION:-

JP INFOTECH
SOFTWARE DEVELOPMENT & RESEARCH DIVISION
www.jpinfotech.org
(0)9952649690

Chennai Office : JP INFOTECH, Old No.31, New No. 86, 1st Floor , 1st Avenue , Ashok Pillar , Chennai - 83.
Landmark : Next to Kotak Mahendra Bank.

Pondicherry Office : JP INFOTECH , #45 , Kamaraj Salai, Thattanchavady, Puducherry
Landmark : Next to VVP Nagar Arch.

jpinfotechprojects@gmail.com

- ✓ Processor - Pentium –IV
- ✓ Speed - 1.1 Ghz
- ✓ RAM - 256 MB(min)
- ✓ Hard Disk - 20 GB
- ✓ Key Board - Standard Windows Keyboard
- ✓ Mouse - Two or Three Button Mouse
- ✓ Monitor - SVGA

SOFTWARE CONFIGURATION:-

- ✓ Operating System : Windows XP
- ✓ Programming Language : JAVA/J2EE
- ✓ Java Version : JDK 1.6 & above.
- ✓ IDE : Netbeans 7.2.1