

## Annotating Search Results from Web Databases

V.Yugandhar, P.T.Sirisha

DEPARTMENT OF MCA

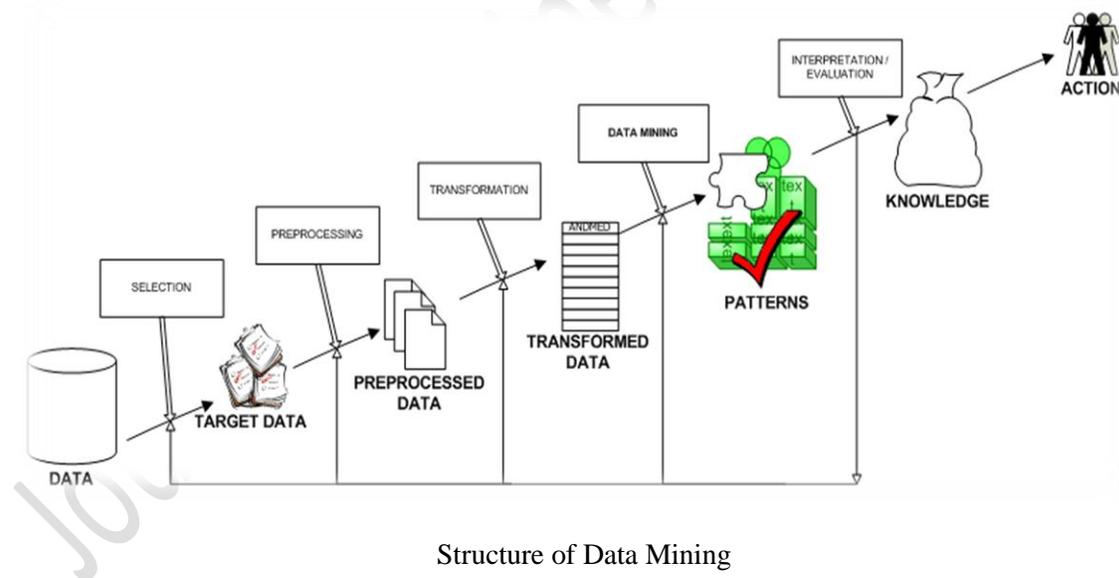
Sri Padmavathi College Of Computer Sciences & Technology

### 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.

### I. INTRODUCTION

#### What is Data Mining?



Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

## How Data Mining Works?

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. **Generally, any of four types of relationships are sought:**

- **Classes:** Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- **Clusters:** Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- **Associations:** Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- **Sequential patterns:** Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

## Data mining consists of five major elements:

- 1) Extract, transform, and load transaction data onto the data warehouse system.
- 2) Store and manage the data in a multidimensional database system.
- 3) Provide data access to business analysts and information technology professionals.
- 4) Analyze the data by application software.
- 5) Present the data in a useful format, such as a graph or table.

## Different levels of analysis are available:

- **Artificial neural networks:** Non-linear predictive models that learn through training and resemble biological neural networks in structure.
- **Genetic algorithms:** Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
- **Decision trees:** Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.

- **Nearest neighbor method:** A technique that classifies each record in a dataset based on a combination of the classes of the  $k$  record(s) most similar to it in a historical dataset (where  $k=1$ ). Sometimes called the  $k$ -nearest neighbor technique.
- **Rule induction:** The extraction of useful if-then rules from data based on statistical significance.
- **Data visualization:** The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

#### Characteristics of Data Mining:

- **Large quantities of data:** The volume of data so great it has to be analyzed by automated techniques e.g. satellite information, credit card transactions etc.
- **Noisy, incomplete data:** Imprecise data is the characteristic of all data collection.
- **Complex data structure:** conventional statistical analysis not possible
- **Heterogeneous data stored in legacy systems**

#### Benefits of Data Mining:

- 1) It's one of the most effective services that are available today. With the help of data mining, one can discover precious information about the customers and their behavior for a specific set of products and evaluate and analyze, store, mine and load data related to them
- 2) An analytical CRM model and strategic business related decisions can be made with the help of data mining as it helps in providing a complete synopsis of customers
- 3) An endless number of organizations have installed data mining projects and it has helped them see their own companies make an unprecedented improvement in their marketing strategies (Campaigns)
- 4) Data mining is generally used by organizations with a solid customer focus. For its flexible nature as far as applicability is concerned is being used vehemently in applications to foresee crucial data including industry analysis and consumer buying behaviors
- 5) Fast paced and prompt access to data along with economic processing techniques have made data mining one of the most suitable services that a company seek

#### Advantages of Data Mining:

1. Marketing / Retail:

Data mining helps marketing companies build models based on historical data to predict who will respond to the new marketing campaigns such as direct mail, online marketing campaign...etc. Through the results, marketers will have appropriate approach to sell profitable products to targeted customers.

Data mining brings a lot of benefits to retail companies in the same way as marketing. Through market basket analysis, a store can have an appropriate production arrangement in a way that customers can buy

frequent buying products together with pleasant. In addition, it also helps the retail companies offer certain discounts for particular products that will attract more customers.

## 2. Finance / Banking

Data mining gives financial institutions information about loan information and credit reporting. By building a model from historical customer's data, the bank and financial institution can determine good and bad loans. In addition, data mining helps banks detect fraudulent credit card transactions to protect credit card's owner.

## 3. Manufacturing

By applying data mining in operational engineering data, manufacturers can detect faulty equipments and determine optimal control parameters. For example semi-conductor manufacturers has a challenge that even the conditions of manufacturing environments at different wafer production plants are similar, the quality of wafer are lot the same and some for unknown reasons even has defects. Data mining has been applying to determine the ranges of control parameters that lead to the production of golden wafer. Then those optimal control parameters are used to manufacture wafers with desired quality.

## 4. Governments

Data mining helps government agency by digging and analyzing records of financial transaction to build patterns that can detect money laundering or criminal activities.

## 5. Law enforcement:

Data mining can aid law enforcers in identifying criminal suspects as well as apprehending these criminals by examining trends in location, crime type, habit, and other patterns of behaviors.

## 6. Researchers:

Data mining can assist researchers by speeding up their data analyzing process; thus, allowing those more time to work on other projects.

## II. 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.

### **III. PROPOSED SYSTEM:**

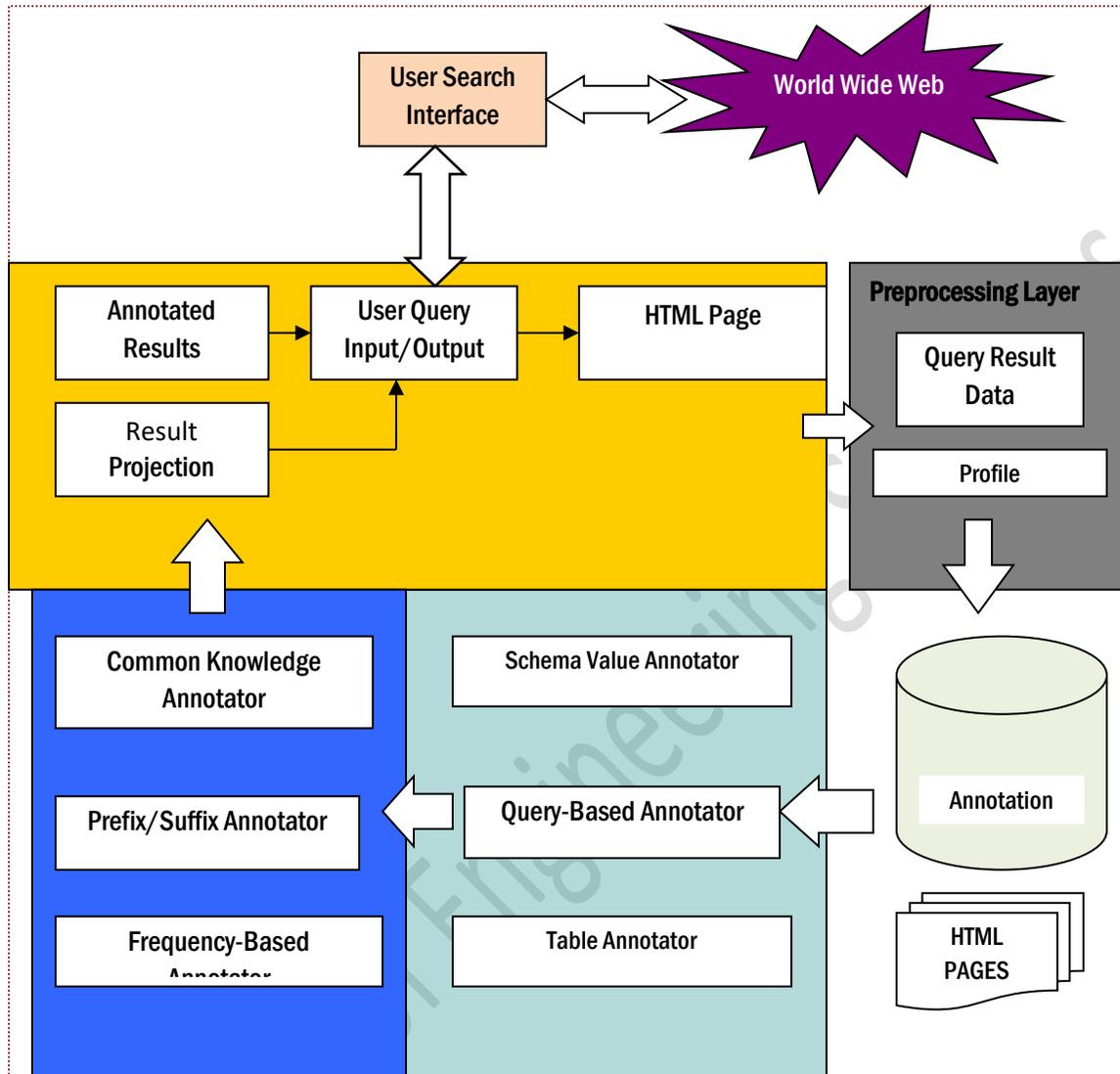
In this paper, we consider how to automatically assign labels to the data units within the SRRs returned from WDBs. Given a set of SRRs that have been extracted from a result page returned from a WDB, our automatic annotation solution consists of three phases.

### **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.

**IV. PROPOSED SYSTEM ARCHITECTURE:**



**V. 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 aWDB are 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  $A_j$  that has the highest matching score among all attributes and then uses  $gn(A_j)$  to annotate the group  $G_i$ . 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.

## **VI. SYSTEM CONFIGURATION:-**

### **HARDWARE CONFIGURATION:-**

- ✓ 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

## **VII. CONCLUSION**

In this paper, we studied the data annotation problem and proposed a multi-annotator approach to automatically constructing an annotation wrapper for annotating the search result records retrieved from any given web database. This approach consists of six basic annotators and a probabilistic method to

combine the basic annotators. Each of these annotators exploits one type of features for annotation and our experimental results show that each of the annotators is useful and they together are capable of generating high quality annotation. A special feature of our method is that, when annotating the results retrieved from a web database, it utilizes both the LIS of the web database and the IIS of multiple web databases in the same domain. We also explained how the use of the IIS can help alleviate the local interface schema inadequacy problem and the inconsistent label problem.

In this paper, we also studied the automatic data alignment problem. Accurate alignment is critical to achieving holistic and accurate annotation. Our method is a clustering based shifting method utilizing richer yet automatically obtainable features. This method is capable of handling a variety of relationships between HTML text nodes and data units, including one-to-one, one-to-many, many-to-one, and one-to-nothing. Our experimental results show that the precision and recall of this method are both above 98 percent. There is still room for improvement in several areas. For example, we need to enhance our method to split composite text node when there are no explicit separators. We would also like to try using different machine learning techniques and using more sample pages from each training site to obtain the feature weights so that we can identify the best technique to the data alignment problem.

#### VIII. REFERENCES

- [1] A. Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages," Proc. SIGMOD Int'l Conf. Management of Data, 2003.
- [2] L. Arlotta, V. Crescenzi, G. Mecca, and P. Merialdo, "Automatic Annotation of Data Extracted from Large Web Sites," Proc. Sixth Int'l Workshop the Web and Databases (WebDB), 2003.
- [3] P. Chan and S. Stolfo, "Experiments on Multistrategy Learning by Meta-Learning," Proc. Second Int'l Conf. Information and Knowledge Management (CIKM), 1993.
- [4] W. Bruce Croft, "Combining Approaches for Information Retrieval," Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval, Kluwer Academic, 2000.
- [5] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRUNNER: Towards Automatic Data Extraction from Large Web Sites," Proc. Very Large Data Bases (VLDB) Conf., 2001.
- [6] S. Dill et al., "SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation," Proc. 12th Int'l Conf. World Wide Web (WWW) Conf., 2003.
- [7] H. Elmeleegy, J. Madhavan, and A. Halevy, "Harvesting Relational Tables from Lists on the Web," Proc. Very Large Databases (VLDB) Conf., 2009.
- [8] D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y. Ng, and R. Smith, "Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages," Data and Knowledge Eng., vol. 31, no. 3, pp. 227-251, 1999.

[9] D. Freitag, "Multistrategy Learning for Information Extraction," Proc. 15th Int'l Conf. Machine Learning (ICML), 1998.

[10] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, 1989.

[11] S. Handschuh, S. Staab, and R. Volz, "On Deep Annotation," Proc. 12th Int'l Conf. World Wide Web (WWW), 2003.

[12] S. Handschuh and S. Staab, "Authoring and Annotation of Web Pages in CREAM," Proc. 11th Int'l Conf. World Wide Web (WWW), 2003

Journal of Engineering Sciences