Abstract:
Most search engines are returning results page that are come from structured databases. This type of search engines also referred as web databases. Each results page contains search results records and each record contains multiple data items. In the proposed method we are introducing 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.

Keywords: Data annotation, Annotation label, Wrapper generation

1. Introduction:
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 describing the relationships between text nodes and data units in detail. Here, 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. Here the ISBNs can be compared to achieve this. 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. Early applications require tremendous human efforts to annotate data units manually, which severely limit their scalability. In the proposed method, we consider how to automatically assign labels to the data units within the SRRs returned from WDBs. Grouping data units of the same semantic can help in identifying the common patterns and features among these data units. These common features are the basis of our annotators. In 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. However, to make the data records and data items in them machine processable, which is needed in many applications such as deep Web crawling and meta searching, the structured data need to be extracted from the deep Web pages. Here, we study the problem of automatically extracting the structured data, including data records and data items, from the deep Web pages. The problem of Web data extraction has received a lot of attention in recent years and most of the proposed solutions are based on analyzing the HTML source code or the tag trees of the Web pages. These solutions have the following main limitations: First, they are Web-page-programming-language dependent, or more precisely, HTML-dependent. As most Web pages are written in HTML, it is not surprising that all previous solutions are based on analyzing the HTML source code of Web pages.

A large portion of the deep web is database-based, i.e., the data encoded in the result pages returned by search engines come from the underlying structured databases (e.g., relational databases). Such type of search engines is usually referred to as Web databases. A typical result page returned from a Web database consists of multiple search result records (SRRs). Usually, one SRR contains multiple data units each of which describes one aspect of the corresponding entity. For example, we are taking a portion of a sample result page returned from a book search engine with three records on it. Each record represents one book and consists of several data units (title, author, etc.). There is a high demand for collecting data of interest from multiple Web databases.

Here, we are describing the DeLa (Data Extraction and Label Assignment) system that sends queries through HTML forms, automatically extracts data objects from the retrieved web pages, fits the extracted data into a table and assigns labels to the attributes of the data objects, i.e., the columns of the table. Our approach is based on two main observations. First, data objects contained in dynamically generated web pages share a common HTML tag structure
and they are listed continuously in the web pages. Based on this, we automatically generate regular expression wrappers to extract such data and fit them into a table. Second, the form contained in a web page, through which users submit their queries, provides a sketch of (part of) the underlying relational database of the web site. Based on this, we extract the labels of the HTML form elements and match them to the columns of the data table, thereby annotating the attributes of the extracted data.

2. Related Work:

Embley et al. utilize ontologies together with several heuristics to automatically extract data in multirecord documents and label them. However, ontologies for different domains must be constructed manually. Mukherjee et al. exploit the presentation styles and the spatial locality of semantically related items, but its learning process for annotation is domain dependent. Moreover, a seed of instances of semantic concepts in a set of HTML documents needs to be hand labeled. These methods are not fully automatic. Arlotta et al. basically annotate data units with the closest labels on result pages. This method has limited applicability because many WDBs do not encode data units with their labels on result pages.

Among all existing researches, DeLa is the most similar to our work. But our approach is significantly different from DeLa’s approach. First, DeLa’s alignment method is purely based on HTML tags, while ours uses other important features such as data type, text content, and adjacency information. Second, our method handles all types of relationships between text nodes and data units, whereas DeLa deals with only two of them (i.e., one-to-one and one-to-many). Third, DeLa and our approach utilize different search interfaces of WDBs for annotation. Ours uses an IIS of multiple WDBs in the same domain, whereas DeLa uses only the local interface schema (LIS) of each individual WDB.

Early approaches to generate wrappers for web sites were mainly based on hand coding, which needs experts to analyze each web site, extract its specific data object structures and manually construct wrappers. Later approaches for wrapper generation are mainly based on unsupervised learning of user labeled examples to extract data objects from web pages. The semantics of the extracted data are given in the user labeled examples.

Another category of Information Extraction systems uses supervised wrapper generation, providing a GUI to interact with users during the wrapper generation process and data labels are provided by users during the interaction.

Literature Survey:

Many web sites contain large collections of pages generated using a common template or layout. For example, Amazon lays out the author, title, comments, etc. in the same way in all the book pages. The values used to generate the pages (e.g., the author, title...) typically come from a database. We have studied the problem of automatically extracting database values from such a collection of web pages automatically without any human input. This page contains the experimental results of applying our techniques to real web page collections. Some of the collections that we used in our experiments were obtained from Road Runner Project which tries to solve a similar problem. The other collections were manually crawled from well-known data-rich sites like E-bay and Netflix.

Data extraction from web pages is performed by software modules called wrappers. Recently, some systems for the automatic generation of wrappers have been proposed in the literature. These systems are based on unsupervised inference techniques taking as input a small set of sample pages, they can produce a common wrapper to extract relevant data. However, due to the automatic nature of the approach, the data extracted by these wrappers have anonymous names. In the framework of our ongoing project RoadRunner, we have developed a prototype, called Labeller that automatically annotates data extracted by automatically generated wrappers. Although Labeller has been developed as a companion system to our wrapper generator, its underlying approach has a general validity and therefore it can be applied together with other wrapper generator systems. We have experimented the prototype over several real-life web sites obtaining encouraging results.

3. Problem Statement:

3.1: 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.
• Early applications require tremendous human efforts to annotate data units manually, which severely limit their scalability.
• 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.

3.2: Proposed System:
• In the proposed system, 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.
• **Phase 1: 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: 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: 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.

Advantages:
• 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.

4. Algorithm evaluation steps:
• **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.

5. System Architecture:

6. Modules Description:
• **Admin**
  - Add URL: In this module url’s and related content which is useful for users is added.
• **User**
  - Searching
    • By URL
    • By Author
• Year
• Title
• Content

• In this module user can search related information by using title or author etc.
• View SRR’s: In this module user can view srr’s in table format.

SRR’S Evaluation:

Query-Based Annotator

The basic idea of this annotator is that the returned SRRs from a WDB 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.

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.

7. Conclusion:

Here, 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 several 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 in alleviating the local interface schema inadequacy problem and the inconsistent label problem.

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.

8. References:


Mrs. Soumya Snigdha Mohapatra is a student of Aditya Engineering College, Surampalem. Presently she is pursuing here M.Tech [CSE] from this college and she received her B.Tech degree in Computer Science and Engineering from C.V.Raman College of engineering affiliated to BPUT Orissa, in the year 2009. Her area of interest includes Data Mining, Object oriented languages and current trends in Computer Science.

Mr. M.Kalyan Ram, well known Author and excellent teacher and Received M.C.A and M.Tech (CSE) from JNTUK and pursuing Ph.D in GITAMU, Visakhapatnam. He is working as Assistant Professor, Department of Computer science engineering, Aditya Engineering College, Surampalem .He has 5 years of teaching experience and couple of publications both international conferences/journals to his credit. His area of Interest includes Data Warehouse and Data Mining.