IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 3, March 2015.
www.ijiset.com
ISSN 2348 – 7968

# An Efficient Content and Query Search for Document Narration

[1]N.Bagyalakshmi, [2]G.V.Sriramakrishnan,M.E,(Ph.D)

[1]Computer science and Engineering, IFET College of Engineering,Tamil Nadu,India

[2]Associate Professor (CSE)

IFET College of Engineering,Tamil Nadu,India.

***Abstract:*** **Annotation process can facilitate subsequent information discovery. Information extraction algorithms facilitate the extraction of structured relations. Author generates a new document and uploads it to the repository. After the upload, analyzes the text and creates an adaptive insertion form. The form contains the best attribute names given the document text and the information need, and the most probable attribute values given the document text.The author can inspect the form, modify the generated metadata as- necessary, and submit the annotated document for storage. Alchemy algorithm can be used to annotate a document. This approach consists of six basic annotators and a probabilistic method to combine the basic annotators in recommended system focus on 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.**

**Keywords -Document Annotation,Query value Annotators,probabilistic method.**

## 1. Introduction:

Online databases, called web databases, comprise the deep 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. Many web applications, such as Meta querying, data integration and comparison shopping, need the data from multiple web databases. 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. 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. Unfortunately, the semantic labels of data units are often not provided in result pages. For instance, 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 for later analysis.

## 2. RECOMMENDER SYSTEMS:

### HTML Parsing

It parses terms from input document. The HTML parser reads the content of a web page into character sequences, and then marks the blocks of HTML tags and the blocks of text content it parses terms from input document. The HTML parser reads the content of a web page into character sequences, and then marks the blocks of HTML tags and the blocks of text content**.** The two fundamental use-cases that are handled by the parser are extraction and transformation While prior versions concentrated on data extraction from web pages, Version 1.4 of the HTMLParser has substantial improvements in the area of transforming web pages, with simplified tag creation and editing, and verbatim toHtml () method output. The program also lets you annotate your tables and fields.

ISSN 2348 – 7968

**Table annotator**

Our Table Annotator works as follows: First, it identifies all the column headers of the table. Second, for each SRR, it takes a data unit in a cell and selects the column header whose area (determined by coordinates) has the maximum vertical overlap (i.e., based on the x-axis) with the cell. This unit is then assigned with this column header and labeled by the header text A (actually by its corresponding global name gn(A) if gn(A) exists). The remaining data units are processed similarly.

**Query-Based Annotator(QA)**

Image annotations and queries are considered as small text-documents that are to be compared. Commonly, a measure based on word matching is used for determining similarity between query and annotations.

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.

**SYSTEM FLOW DIAGRAM**


**Schema Value Annotator (SA)**

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.

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 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 matched query data. In personalized search, this search is done relative to the field problem to the use.

**FUTURE WORK**

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 as mentioned in Section. 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 in the feature.

**CONCLUSION**

The data annotation problem and proposed a multiannotator 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