Annotating search using Web databases

Poonam V. Wankhede[1]
Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad (MS), India

Sachin N. Deshmukh[2]
Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad (MS), India

***Abstract*— For many search engines, data encoded in the returned result pages come from the underlying structured databases i.e Deep web is a database based. Such type of search engines is often referred as Web databases (WDB). A web database contain a typical many search results records. Each SRR contain multiple data units which need to be label semantically for machine processable. Early applications require tremendous human efforts to annotate data units manually, which severely limit their scalability. Now we present an automatic annotation approach which contains the data units on the web result page into a different groups such that same groups have the same semantic labels. Then the six annotations are combined and predict the final annotation label. The last is the wrapper generation, with the help of wrapper generation we annotate the new result page from the same web database. Our results contain precision and recall.**

***Index Terms*— Data alignment, data annotation, web database, wrapper generation**

I. INTRODUCTION

Now a day a large portion of the database is a Deep Web database i.e data encoded in the returned result pages of many search engines come from the underlying structured databases. The search engines are referred to the Web database. A Web database contains a multiple search result records and each search result record corresponds to an entity. For example we consider a Web page that contain a information about a book and normally the each search result contain a multiple data units or instances like book title, book name, book publication, book price and so on. Unfortunately not all data units are encoded with meaningful labels. Sometimes the result labeled with title and publication the users not recognized it easily. So we propose how to automatically annotate the data units in the SRRs returned by Web databases and assigning meaningful labels to them.

Due to the rapid growth of the deep Web and multiple web database, annotation problem have become very important. For the data analysis and mining we have to collect data from multiple Web database, the most important thing is that the data units are correctly labeled so they can organized appropriately and stored for subsequent machine processing. Web services interfaces in the search sites, it may be easier to annotate their SRRs because the semantic meanings of their data units are more clearly described in Web Services Description Language (WSDL). However, our investigation indicates that very few search sites have Web services interfaces. One reason for this phenomenon may be that Web services are mainly designed to support business to business (B2B) applications while most search sites are for business to customer (B2C) applications. Therefore, it is necessary to extract and annotate data from the legacy HTML pages. Here we are propose a data level annotation.

II. LITERATURE SURVEY

W. Liu, X. Meng, and W. Meng et al.[1] design a system for extracting structured data from deep Web pages is a challenging problem due to the underlying complex structures of such pages. A large number of techniques have been proposed to address this problem, but all of them have intrinsic limitations because they are Web-page programming- language-dependent. This approach primarily utilizes the visual features on the deep Web pages to implement deep Web data extraction, including data record extraction and data item extraction. It is also proposed as new evaluation measure revision to capture the amount of human effort needed to produce perfect extraction.

J. Madhavan, D. Ko, L. Lot, V. Ganapathy et al[2] design a system for hiding content behind HTML forms has long been acknowledged as a significant gap in search engine coverage. The system developing Deep Web content, i.e., pre-computing submission for the HTML forms and inserting the resulting HTML pages into a search engine index. The results of our developing have been incorporated into the Google search engine and today drive more than a thousand queries per second to Deep-Web content.

S. Mukherjee, I.V. Ramakrishnan, and A. Singh et al[3] develop a technique for identifying and annotating semantic concepts implicit in such documents makes them directly open to for Semantic Web processing. The paper describes a highly automated technique for annotating HTML documents, especially template-based content-rich documents, containing many different semantic concepts per document. Starting with a (small) seed of hand-labeled instances of semantic concepts in a set of HTML documents we bootstrap (starting of a self-sustaining process that is supposed to proceed without external input) an annotation process that automatically identifies unlabeled concept instances present in other documents. The bootstrapping

technique overwork the observation that semantically related items in content rich documents exhibit consistency in presentation style and spatial locality to learn a statistical model for accurately identifying different semantic concepts in HTML documents drawn from a variety of Web sources. We also present experimental results on the effectiveness of the technique.

Y. Zhai and B. Liu et al[4] design a system for problem of extracting data from a Web page that contains several structured data records. Extracting data from a Web page the first class of methods is based on machine learning, which requires human labeling of many examples from each Web site that one is interested in extracting data from. This process is more time consuming due to large number of sites and pages on the Web. The next second class method is based on automatic pattern discovery These methods are either inaccurate or make many assumptions.

In recent years Web information extraction and annotation is an active area. A lot of system like wrapper induction system have confident and faith on human [5] [6] to generate the wrapper on the marked data of the sample page. Because of some supervised training and learning process, these systems can achieve high extraction accuracy. But it gives a result of poor scalability for the application that need to extract information from large number of web source.

III. EXPERIMENTAL WORK

We are design a system which contain the information related to book which developed online and offline. We take three search sites for book which are Flipkart, booksamillion, powells. With the help of this three book sites we retrieve the web pages related to book and store it. This web pages we have to consider our dataset. For the extraction of the data unit and text node we use a five different feature. First is Data Content, second is Presentation Style third is Data-type, forth is Tag Path and last fifth is Adjacency.

Data content

The data units or text nodes share certain keywords with the same ideas. The first, user enter a searching statement that usually contain the search keywords which is matching to the search field. Consider an example the sample Web pages return for the multiple search field with a particular keyword for ex Author. Second, when for good understanding of data content we have to put labels in front of that certain data. In our system design for book application so price of every book has leading words “Our Price” in the same text node[7].

Presentation Style

Presentation style describes how a data unit is displayed on a webpage. It consists of six presentation style features: font face, font size, font color, font weight, text decoration (underline, strike, etc.), and whether it is italic. Different SRRs Data units of the same concept in are usually displayed in the same style. For example, all the availability information is displayed in the exactly same presentation style.

Data Type

In the HTML code all the data unit has its own semantic type. The subsequent basic data types are currently considered in our approach: Date, Time, Currency, Integer, Decimal, Percentage, Symbol, and String. Further String type is defined in All-Capitalized-String, First- Letter-Capitalized-String, and Ordinary String.

Tag Path

We use ViNTs for SRR, the sequence of tags traversing from root of the SRR to the corresponding node in the tag tree [8]. All the node represent the two paths in which one is the tag name and other is the direction whether the next node is the next sibling is denoted by “S” and the first child denoted by the “C”.

Adjacency

For the data adjacency we can consider a given data unit d in the SRR, then next we have to consider d in the SRR having the data units d_p and d_s immediately before and after d. d_p and d_s are the preceding and succeeding data units of d, respectively. D_1 and d_2 are the two data unit but they are in separate SRR. If d_{p1} and d_{p2} are belong the same concept and the d_{s1} and d_{s2} are also belonging to the same concept the we can easily say that d_1 and d_2 are also belonging to the same concept.

The next we have to use six basic annotators

- **Table Annotator (TA)**
- **Query-Based Annotator (QA)**
- **Schema Value Annotator (SA)**
- **Frequency-Based Annotator (FA)**
- **In-Text Prefix/Suffix Annotator (IA)**
- **Common Knowledge Annotator (CA)**

Table Annotator:

The first annotator is table based annotator, many database to organized data units are in the table format. The table contain header which is presented in the top of the table. The table stored the data in the column and the row manner. The header is used to represent the meaning of each column and row is used to represent search result request data.

Query based annotator:

The second annotator is a query based annotator, in which user enter a query related to that data the result will be display and then user query and the data units are compared for the required column. The result is display related to that query.

Schema Value-based Annotator:

The third is a schema value based annotator, many attributes in a schema value on the search interface is predefined values. for example book related query and the related attribute authors it may have a set of predefined values i.e authors in that list. . If the group having several data units, the Schema Value-based Annotator is used to find out the best synchronized attribute to the group from the IIS. The schema that firstly discovers that the uppermost matching score among all the attribute and then it annotate the group.

Text Frequency-Based Annotator:

The fourth is the text frequency based annotator, SRR contain the records in the result page. The grouping of data is depend on the present content of that data. Same data are in one group and similarly the same for next group of data and so on. some

data have low frequency and some of have high frequency. the data unit of higher frequency is their attribute name and the data of lower frequency is their values. For this calculation compute the cosine similarity between the attribute and the data unit. Text Frequency-Based Annotator found the general preceding units shared by all the data units of the group. The data units with the superior frequency are plausible attribute name. And the data units with the low frequency are most likely appear from databases as values.

Prefix/Suffix Annotator:
The fifth basic annotator is the prefix and suffix annotator, each search result contain a multiple search result record in the web result page. This result page also contain some prefix and the some suffix with them. For $ contain related to price so it come before price value. The prefix and suffix annotator is to check all the data units have same prefix or suffix, if it is match then it used to annotate the data units inside the next group.

Common Knowledge Annotator:
The six and the last annotator is the common knowledge, when we searching online product and the multiple results are showing to us. Then it show some related information that product, the product buy or not because it shows "in the stock" for availability and "Out of the stock" means the book not available right now, this identifications for the human because it is common thing to remember.
We use alignment algorithm for align our data and extraction algorithm for extracting HTML pages.

Extraction algorithm

We extract Book Title, Book Price, Book Author name, Book Publisher name, Book Publication year etc.
1. Given input as a HTML page
2. It identify the features(Title,Authorname,price,Publication year)
3. Extract Title by giving class name(doc.getByclassname("name of class"))
4. For i ← 1 to no of SRR
5. res[i] ← SRR[i]
6. Forj← 1 to res.length
7. data[row][column] ←rse[j]
8. Column++
9. Extract Author Name
10. Fori← 1 to no of SRR
11. res[i] ← SRR[i]
12. For j ← 1 to res.length
13. data[row][column]←rse[j]
14. column++
15. End

Alignment Algorithm

1. j← 1;column ← 0;
2. While true \\Create alignment groups
3. for i←1 to no of SRR
4. Gj [i][j]
5. If Gj is empty
6. exit \\break the loop
7. V (G)
8. If |V |>1 \\Collect all data units in one group
9. Forx ← 1 to no of SRR
10. Fory ← 1 to no of SRR[i].length
11. table[y][column][x][y]

Shifting

12. Fork ← 1 to |V |
13. For each SRR[x][j] in V[k]
14. Insert null at position j in SRR[x]
15. j←j+1
16.column←column+1
17.continue this process for all table columns.

IV. RESULT

Precision and recall are the basic measures used in evaluating search strategies. Precision and recall both are inversely proportional to each other.
A =No of relevant records retrieved.
B = No of relevant records not retrieved
C = No of irrelevant records retrieved

PRECISION
Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage.

$$\text{Precision} = \frac{\text{No. of relevant records retrieved}}{(\text{No. of relevant records retrieved} + \text{No of irrelevant records retrieved})} * 100 \quad (1)$$

RECALL
Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.

$$\text{Recall} = \frac{\text{No. of relevant records retrieved}}{(\text{No. of relevant records retrieved} + \text{No of relevant records not retrieved})} * 100 \quad (2)$$

ACCURACY
The degree to which the result of measurement calculations or specification confoms to correct value or standard.

TABLE 1: Precision, Recall and Accuracy Calculations:

Domain	Book Name	WebSite Name	Precision	Recall	Accuracy
Book					
	Java	Flipkart	80	64	85
	Java	BooksAmillion	78	65	85
	Java	Powells	74	65	85

TABLE 2:Precision, Recall and Accuracy Calculations:

Domain	Book Name	WebSite Name	Precision	Recall	Accuracy
Book					
	Operating System	Flipkart	70	62	75
	Operating System	BooksAmillion	76	64	83
	Operating System	Powells	76	61	84

TABLE 3: Precision, Recall and Accuracy Calculations:

Domain	Book Name	WebSite Name	Precision	Recall	Accuracy
Book					
	Computer Network	Flipkart	75	65	80
	Computer Network	BooksAmillion	76	65	85
	Computer Network	Powells	78	64	88

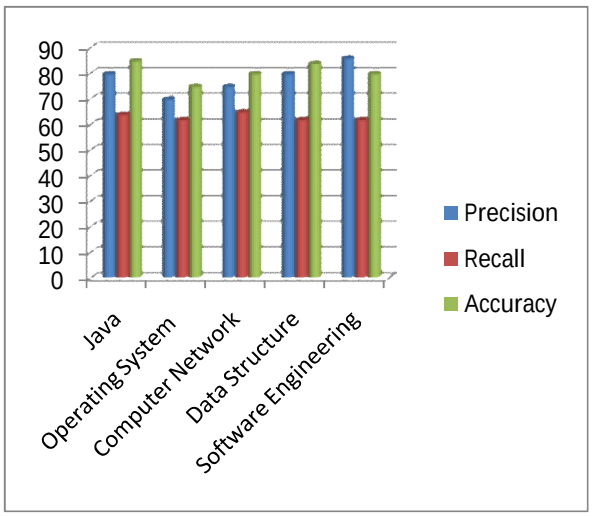

Fig 1: Flipkart Precision, Recall and Accuracy Graphical Results

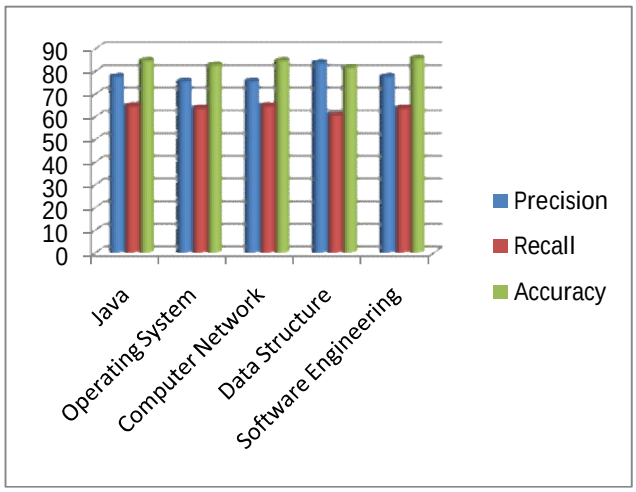

Fig 2: Booksamillion Precision, Recall and Accuracy Graphical Results

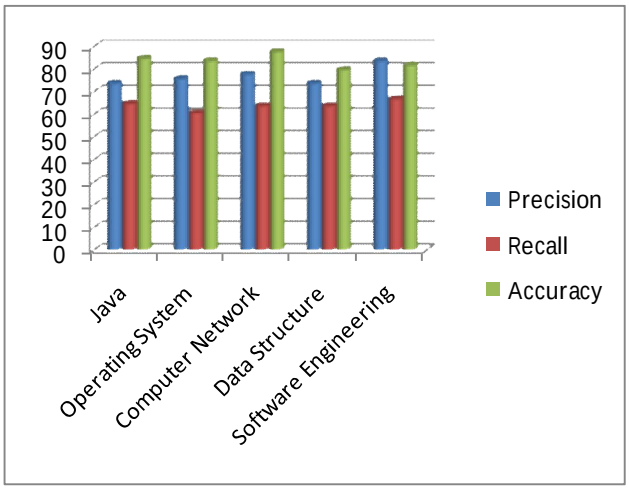

Fig 3: Powells Precision, Recall and Accuracy Graphical Results

Performance of Annotators

Here we lists our performance of basic annotators . We can see that an average precision and recall are high which show our annotation method is effective. Our annotation method is domain independent because they give the high precision and recall of each domain.

TABLE 4: Annotators

Domain	Precision	Recall
Table Annotator	40%	30%
Query Based Annotator	80%	60%
Schema Value Annotator	50%	40%
Frequency Based Annotator	76%	62%
Infix/suffix Annotator	70%	62%
Common Knowledge Annotator	80%	60%
Avg	66%	52%

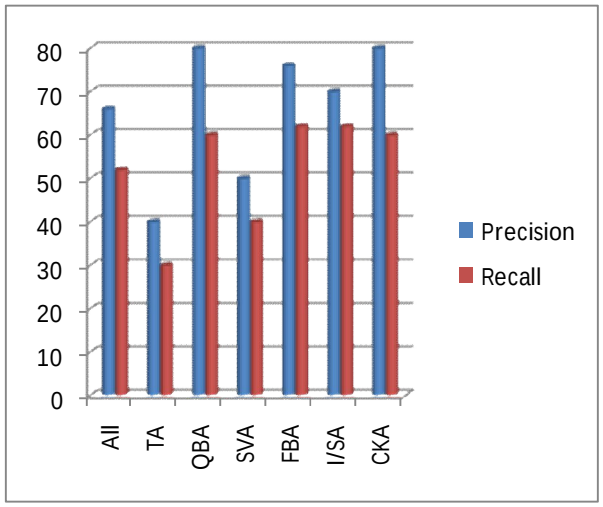

Fig 4: Evaluation of basic annotators

V. CONCLUSION

An annotation is a great need of the current situation to enhance information search with different types of annotations. 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 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.

VI. ACKNOWLEDGMENT

We would like to thanks Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University.

VII. 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] L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources," Proc. IEEE 16th Int'l Conf. Data Eng. (ICDE), 2001.
[6] N.Kruhmerick, D. Weld, and R.Do orenbos, "Wrapper Induction for Information Extraction", Proc Int'1 joint conf.Artificial Intelligence (IJCAI), 1997.
[7] M.Yazhmozhi1, M. Lavanya2, Dr. N. Rajkumar3 "Annotating Multiple Web Databases Using Svm" International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)Organized by Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014.
[8] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, "FullyAutomatic Wrapper Generation for Search Engines," Proc. Int'lConf. World Wide Web (WWW), 2005.

