IPHV7I3003X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

AUTOMATIC DATA INTEGRATION OF SEARCH INTERFACES ON WEB DATABASES

[#1]Kapapoori Naresh Kumar - M.Tech Pursuing,
[#2]B.Srinivas - Assistant Professor,
Department of Computer Science & Engineering,

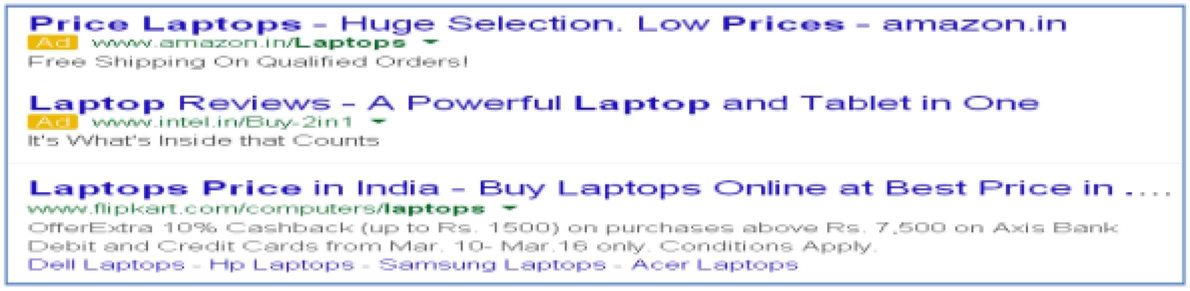
MOTHER THERESSA COLLEGE OF ENGINEERING & TECHNOLOGY, Peddapalli, Karimnagar, TS, India.

ABSTRACT: Internet has paved the way for accessing huge amount of data from across the globe. World Wide Web (WWW) is playing an important role in helping users to obtain required data in various domains. People of all walks of life are using search engines in order to search for data. The results returned by search engines are in the form of web pages that hold results obtained from underlying databases. The search results can be used further in many applications such as data collection, comparison of prices and so on. However, to make these applications successful the search results are to be made machine processable. In order to make them machine processable, it is important that the result pages are annotated in a meaningful fashion. The process of annotating has to consider groups of data and obtain final annotation after aggregating them. Recently Lu et al. proposed various annotators that help in annotating search results. In this paper we built a prototype application that will provide graphical means of making annotations and visualizing them. We implemented one of the methods proposed by Lu et al. in the proposed application. The empirical results revealed that the application is effecting in annotating search results.

KEYWORDS: World Wide Web, search results, annotating search results.

I. INTRODUCTION

The Internet super highway is widely used as a vehicle to information sharing across the globe. People across the globe, of all walks of life, are accessing Internet resources through search engines. The search engines provide web based interface for information search. Search engines return huge amount of data which is presented in encoded format through web pages. However, the data comes from underlying database. The search results from web databases that can be used further in applications like price comparison, data collection, and other related applications. When the search word "LAP TOP PRICES" is given in Google, it returned 24,60,00,000 result pages. Some of the results obtained are as follows.


Price Laptops - Huge Selection. Low Prices - amazon.in
Ad www.amazon.in/Laptops
Free Shipping On Qualified Orders!

Laptop Reviews - A Powerful Laptop and Tablet in One
Ad www.intel.in/Buy-2in1
It's What's Inside that Counts

Laptops Price in India - Buy Laptops Online at Best Price in ....
www.flipkart.com/computers/laptops
OfferExtra 10% Cashback (up to Rs. 1500) on purchases above Rs. 7,500 on Axis Bank Debit and Credit Cards from Mar. 10- Mar.16 only. Conditions Apply.
Dell Laptops - Hp Laptops - Samsung Laptops - Acer Laptops

Figure 1 –Some of the search results

As can be seen in Figure 1, it is evident that there are many search results that are associated with different web pages. The URL associated with each search result is different and the results came from many underlying databases in the web. The search results are to be made machine processable in order to use them further in real world applications. With the annotations, it is possible to process the web pages returned by search engines. For instance, the prices of various companies pertaining to a product can be compared. The comparison web sites can exist that derive data from across the pages returned by search engines. By providing price comparison, the web sites over Internet can help netizens to make well informed decisions. With many processing techniques, the search engines are presenting the results in meaningful way. Earlier the case was different. The results needed much human effort in order to annotate it manually. Recently Lu et al. [1]presented various ways of annotating the search results. They developed a mechanism that will automatically annotate the search results getting rid of manual labeling of web pages. Their solution contains three phases. They are illustrated in Figure 2.

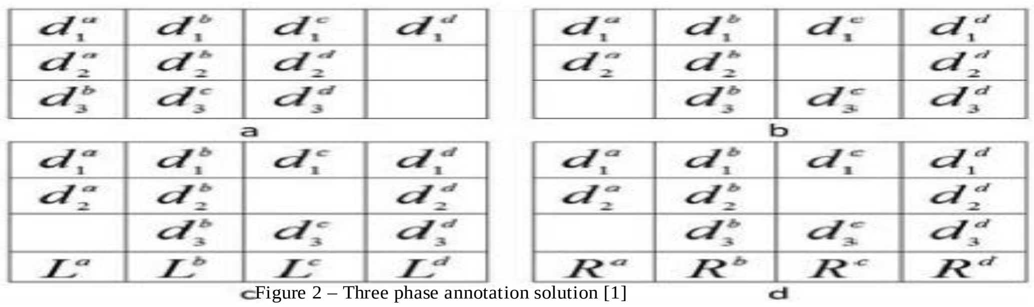


Figure 2 – Three phase annotation solution [1]

The first phase is known as alignment phase where data units are organized into groups based on different concepts. The phase 2 is known as annotation phase which takes care of making annotators that annotate web documents automatically. The phase 3 is known as annotation wrapper generation phase where an annotation rule is generated for each identified concept. Annotation wrapper is the collection of all the rules for all groups which have been aligned. Annotation wrappers help improve the process of annotation. A clustering based scripting technique is used to achieve this.

In this paper we implemented few annotators presented by Lu et al. [1]. We built a prototype application that takes care of automatic annotations of search results. The research results are obtained through Google search. The results are automatically annotated using the mechanism proposed in [1]. The empirical results revealed that our prototype is useful and can be used in the real world. The remainder of the paper is structured as follows. Section II reviews literature that focuses on the prior work pertaining to annotation of search results. Section III presents the proposed approach to achieve automatic annotation of web search results. Section IV presents experimental results while section V concludes the paper.

II. RELATED WORK

Extracting information from web and annotating search results for further processing has been around for some years. This is because there is an important utility in the real world when search results are annotated. Many existing systems that came into existence have manual system for annotating search results. For instance in [2] and [3], human users are involved for marking the annotations. These systems are manual and they are not scalable. However, they achieved high rate of accuracy. Their problemis that they are not scalable and thus can't be used in real world applications [4], [5]. Spatial locality and presentation styles are used in [6] for annotations. However, the process of annotations in this approach is dependent on domains. Ontologism were used in [7] where labeling documents was done based on certain heuristics. Many prior works focused on constructions of wrappers. However, those wrappers could only extract data but not annotations. Many other researches came into existence that focused on automatic allocation of labels to search results [8], [9], and [10].

In [10] data units were annotated with closest labels but the method was not impressive for web databases. Query interfaces were used in [9] and ontologies are constructedin a domain dependent way. For the first time HTML tags are used by DeLa [8] in order to align data units. It was achieved using heuristics. Labeling and attribute extraction is done simultaneously in [11]. In this approach label sets are pre-defined and thus it is not so dynamic. HTML tag paths are the frequently used feature [12]. Visual features are also used in [13] for aligning data. However, it was successful only for text nodes. In [14] a record is split into various segments for data alignment and annotations.

Recently Lu et al. [1] proposed an approach for automatic annotations of search results. First of all their approach considers various kinds of relationships in the data units and handles them. However, the existing works considers only some types as explored in [8] and [6]. Afterwards, Lu et al. used the features together besides ontologyin order to align data. Clustering based scripting algorithm is also used to achieve this. The work in [1] and that in [8] are similar. Both approaches make use of HTML tags for processing and handle all kinds of relationships. However, their approach is different for annotating search results. An annotation wrapper was constructed that can describe rules for assigning labels to search results. Crawling deep web is one of the applications of the annotations. ViNTs [15] was used to obtain records from search results. The previous paper [16] is the basis for the work done by Lu et al. [1].

III.PROPSED SYSTEM FOR ANNOTATING SEARCH RESULTS

In this paper we take the concepts for innovatingsearch results from [1]. Reader can get more basic information from [1]. However, in this section we provide the implementation details of our application and algorithm for automatic annotation of search results. As described in [1], our approach also has three phases in the application. The three phases and their functionality areproviding in the schematic representation as shown in Figure 3.

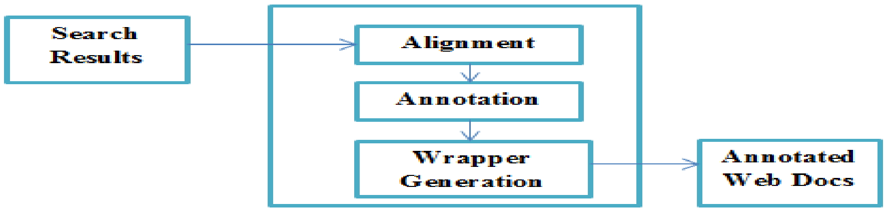
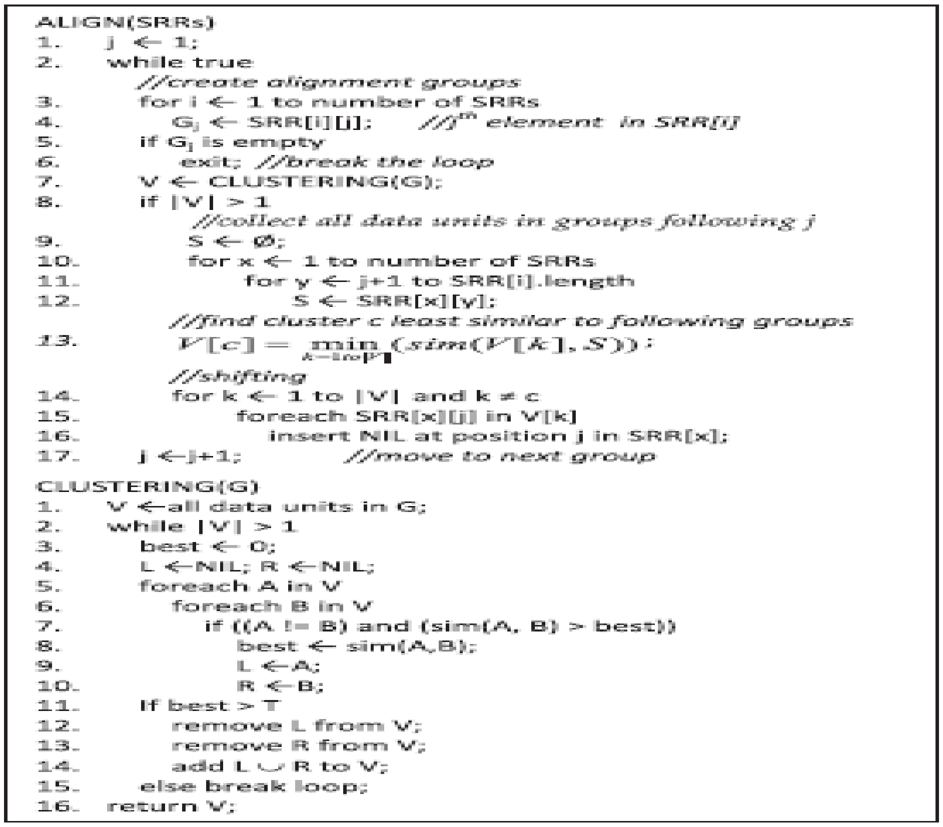


Figure 3 –Proposed framework for automatic annotation

As can be seen in Figure 3, it is evident that the web documents which are search results (taken from Google) are given as input to the system. Then the searchresults are processed in the first phase known as alignment to divide the data into groups and then annotation takes place in the second phase while the third phase focuses on annotation wrappers that provide final annotated web pages. Two kinds of annotators are applied in the proposed prototype application. They are Table Annotator (TA) and Query Based Annotator (QA).

Table Annotator
Many search engines present some data in tabular format. It does mean the search results are presented in tabular format. The data in tabular format can help users to understand it by a glance. The table annotator identified column headers in the table. Afterwards, the data items are processed. The maximum vertical overlap in a column is identified and then the header text is used for labeling.

Query – Based Annotator
This annotator takes the idea that the search results of a query are related to that query. Name of the search field title is used to annotate. A query with multiple query terms, that are pertaining to specific attribute returns records that satisfy the search results. The search results do not have all the attributes that are present in database. For this reason query based annotator is useful in this context.



```
ALIGN(SRRs)
1.    j ← 1;
2.    while true
         //create alignment groups
3.       for i ← 1 to number of SRRs
4.          G_j ← SRR[i][j];     //j^th element in SRR[i]
5.       if G_j is empty
6.          exit; //break the loop
7.       V ← CLUSTERING(G);
8.       if |V| > 1
            //collect all data units in groups following j
9.          S ← Ø;
10.         for x ← 1 to number of SRRs
11.            for y ← j+1 to SRR[i].length
12.               S ← SRR[x][y];
          //find cluster c least similar to following groups
13.         $V[c] = \min_{k=1 \text{ to } |V|}(sim(V[k], S))$;
          //shifting
14.         for k ← 1 to |V| and k ≠ c
15.            foreach SRR[x][j] in V[k]
16.               insert NIL at position j in SRR[x];
17.      j ←j+1;          //move to next group

CLUSTERING(G)
1.    V ←all data units in G;
2.    while |V| > 1
3.       best ← 0;
4.       L ←NIL; R ←NIL;
5.       foreach A in V
6.          foreach B in V
7.             if ((A != B) and (sim(A, B) > best))
8.                best ← sim(A,B);
9.                L ←A;
10.               R ←B;
11.      If best > T
12.         remove L from V;
13.         remove R from V;
14.         add L ∪ R to V;
15.      else break loop;
16.   return V;
```

Figure 4 –Data alignment algorithm [1]

As can be seen in Figure 4, it is evident that the algorithm takes search results as input and generates many clusters or groups that are the result of the alignment process. These groups are used for further processing as illustrated in Figure 1. After the process of alignment, the application will automatically make annotations which are visualized in the prototype application.

Data Alignment Algorithm

The algorithm for data alignment [1] assumes that the attributes of the data are in some specific order for all the rows. The assumptions make the algorithm work in that fashion. Generally this assumption is true for many search results that are presented in tabular format. Figure 4 shows the algorithm that is meant for data alignment.

Prototype Application

The prototype application is built using My Eclipse IDE. Java is the programming language used. The interface provides search facility and the results are used as input to the algorithm. The application is with GUI that makes the application intuitive besides having the capabilities to visualize the results of annotations. The application facilitates saving of results as well for future retrieval and revisions. We used local database for storing results. MY SQL is used as backend in order to store local content. The application has provisions to produce summary of results in graphical format with the help of graphs which are presented in the ensuing section.

IV.EXPERIMENTAL RESUTLS

We have made experiments data from various domains with respect to two annotators only. The annotators used include table annotator and query – based annotator. Both the annotators are supported by the prototype application and it is extensible so as to support more annotators in future. The performance of data alignment and annotation are presented in Table 1.

Domain	Data Alignment Performance		Annotation Performance		Annotation with Wrapper	
	Precision	Recall	Precision	Recall	Precision	Recall
Auto	97.2%	97.4%	96.3%	96.6%	93.5%	90.2%
Book	97.1%	96.2%	96.2%	95.4%	92.6%	91.3%

Table 1 –Experimental results

As presented in Table 1, it is evident that more than 90% precision and recall were recorded for both the performances such as data alignment and annotations. The table also shows the performance of annotation with wrapper. The results are presented in the following graphs.

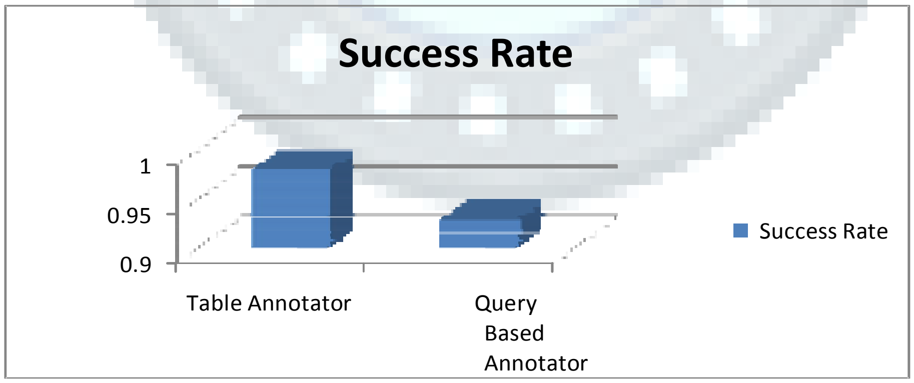


Figure 5– Success rate for table annotator and query based annotator

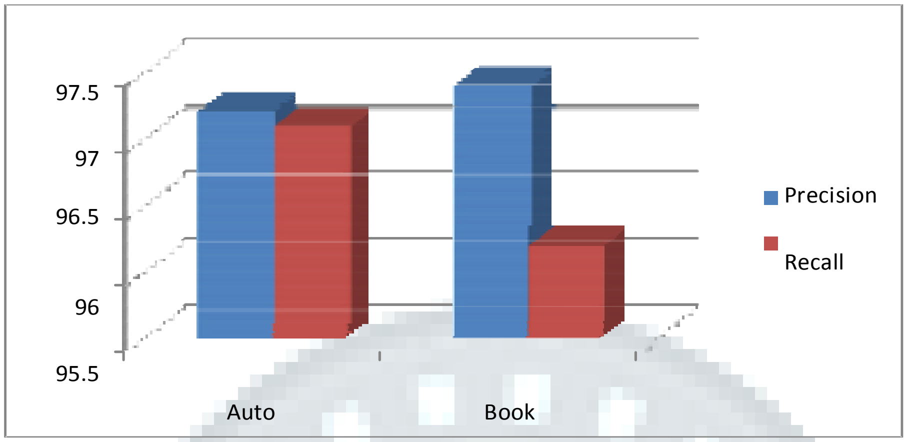


Figure 6 –Performance of data alignment for two domains

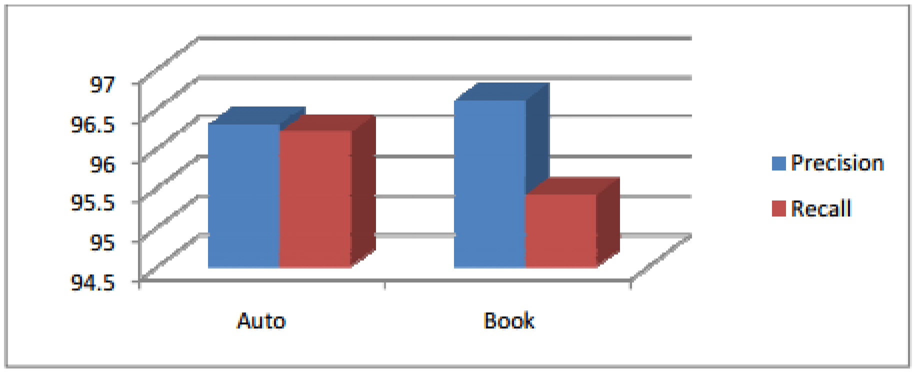


Figure 7 – Performance of data annotation for two domains

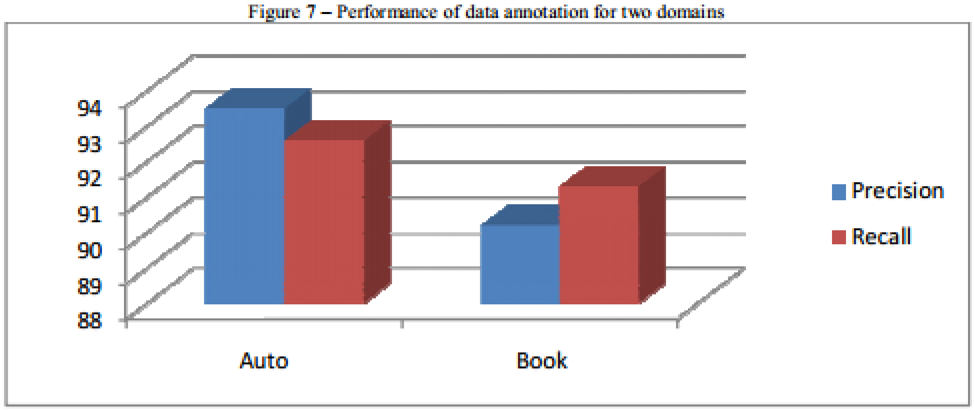


Figure 8 – Performance of data annotation with wrappers for two domains

As shown in Figure 5, Figure 6, and Figure 7, it is evident that the prototype application is capable of producing annotations automatically given search results of Google. The performance of the application is encouraging and the application can be used in the real world applications.

V. CONCLUSION

In this paper we focused on the problem of annotating search results. The search results of search engines form web databases which can be used for further processing in order to leverage them in various applications like content comparison, data extraction and so on. We built a prototype application that facilitatesusers to give a query, and then the query is programmatically submitted to Google. The results of Google are used in the application for further processing. As explored in Figure 1, the three phases are carried out. The phases are alignment phase, annotation phase and wrapper generation phase. After completion of these phases, the application visualizes results which are nothing but the annotated documents. HTML tags are used to process the pages while annotating them. The annotated results are further useful in real world applications. The empirical results revealed that our application is effective.

REFERENCES

[1] Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng and Clement Yu, (2013). Annotating Search Results from Web Databases. IEEE Transactions On Knowledge And Data Engineering, Vol. 25, NO. 3.p1-14.
[2] N. Krushmerick, D. Weld, and R. Doorenbos, “Wrapper Inductionfor Information Extraction,” Proc. Int’l Joint Conf. ArtificialIntelligence (IJCAI), 1997.
[3] L. Liu, C. Pu, and W. Han, “XWRAP: An XML-Enabled WrapperConstruction System for Web Information Sources,” Proc. IEEE16th Int’l Conf. Data Eng. (ICDE), 2001.

[4] Z. Wu et al., “Towards Automatic Incorporation of Search Enginesinto a Large-Scale Metasearch Engine,” Proc. IEEE/WIC Int’l Conf.Web Intelligence (WI ’03), 2003.
[5] W. Meng, C. Yu, and K. Liu, “Building Efficient and EffectiveMetasearch Engines,” ACM Computing Surveys, vol. 34, no. 1,pp. 48-89, 2002. [6] S. Mukherjee, I.V. Ramakrishnan, and A. Singh, “BootstrappingSemantic Annotation for Content-Rich HTML Documents,” Proc.IEEE Int’l Conf. Data Eng. (ICDE), 2005.
[7] D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y. Ng,and R. Smith, “Conceptual-Model-Based Data Extraction fromMultiple-Record Web Pages,” Data and Knowledge Eng., vol. 31,no. 3, pp. 227-251, 1999.
[8] J. Wang and F.H. Lochovsky, “Data Extraction and LabelAssignment for Web Databases,” Proc. 12th Int’l Conf. World WideWeb (WWW), 2003.
[9] W. Su, J. Wang, and F.H. Lochovsky, “ODE: Ontology-AssistedData Extraction,” ACM Trans. Database Systems, vol. 34, no. 2,article 12, June 2009.
[10] L. Arlotta, V. Crescenzi, G. Mecca, and P. Merialdo, “AutomaticAnnotation of Data Extracted from Large Web Sites,” Proc. SixthInt’l Workshop the Web and Databases (WebDB), 2003.
[11] J. Zhu, Z. Nie, J. Wen, B. Zhang, and W.-Y. Ma, “SimultaneousRecord Detection and Attribute Labeling in Web Data Extraction,”Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and DataMining, 2006.
[12] Y. Zhai and B. Liu, “Web Data Extraction Based on Partial TreeAlignment,” Proc. 14th Int’l Conf. World Wide Web (WWW ’05),2005.
[13] W. Liu, X. Meng, and W. Meng, “ViDE: A Vision-Based Approachfor Deep Web Data Extraction,” IEEE Trans. Knowledge and DataEng., vol. 22, no. 3, pp. 447-460, Mar. 2010.
[14] H. Elmeleegy, J. Madhavan, and A. Halevy, “HarvestingRelational Tables from Lists on the Web,” Proc. Very LargeDatabases (VLDB) Conf., 2009.
[15] 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.
[16] Y. Lu, H. He, H. Zhao, W. Meng, and C. Yu, “AnnotatingStructured Data of the Deep Web,” Proc. IEEE 23rd Int’l Conf. DataEng. (ICDE), 2007.