Data Extraction from Deep Web Pages

Jufeng Yang Guangshun Shi Yan Zheng Qingren Wang
Institute of Machine Intelligence
Nankai University, 300071 Tianjin, China
richard_nkimi@yahoo.com.cn {gssshi, yanzheng, qrwang}@imi.nankai.edu.cn

Abstract

In this paper, we propose a novel model to extract data from Deep Web pages. The model has four layers, among which the access schedule, extraction layer and data cleaner are based on the rules of structure, logic and application. In the experiment section, we apply the new model to three intelligent system, scientific paper retrieval, electronic ticket ordering and resume searching. The results show that the proposed method is robust and feasible.

1. Introduction

Most of the information on Internet can not be directly accessed via the static link. Users must type some keywords before getting the information hidden in the Web database. Deep Web contains 400-500 times more information and 15% larger visit capacity than that of Surface Web. Moreover, the quality of data is also relatively higher [1].

The research on Data Extraction from Deep Web pages is becoming a hot area. The study is to help people access automatically and use freely the information distributed on the Deep Web. [2] and [3] focus on the technology of the search of Deep Web database on Internet; [4], [5] and [6] study the methods of analyzing and extracting the attribute from the query interface and the ones of building a uniform pattern, which contribute to integrate several Deep Web. Data extraction is another important aspect of Deep Web research, which involves in extracting the information that users are interested in from semi-structured or unstructured Web pages and saving the information as the XML document or relationship model. [7], [8] and [9] have done a lot of work in this field. Additionally, in some papers, such as [10] and [11], researchers have paid more attention to the influence of semantic information on Deep Web.

The research discussed above covers the key components of the Deep Web Data Extraction. They mainly focus on the design of theoretical models and the ideal methods, while neglecting of the difficulties in trying to apply to practice. The difficulties include: how to choose the driven crawler to get the ideal set of target sites; how to optimize existing analytical methods to be feasible and robust when applied in the pages that have various structures and may be built using different techniques; and how to deal with the Deep Web data and apply them to practical application more appropriately.

In this paper, we propose a robust system model of Deep Web Data Extraction that applies in some specific application areas. It firstly uses the two regularities of the domain knowledge and interface similarity to assign the tasks that are proposed from users, and chooses the most effective set of sites to visit. Then, the model extracts and corrects the data based on logical rules and structural characteristics of acquired pages. Finally, it cleans and orders the raw data set to adapt the customs of application layer for later use by its interface.

The paper is organized as following: Section 2 outlines our approach; Section 3, 4 and 5 describe in detail the proposed algorithm; experiment and results are shown in Section 6 and Section 7 concludes the paper.

2. Approach Overview

2.1 System Model

Based on the previous works, we propose a novel system model which contains four main levels from bottom to top:

Data Link Layer: it is at the lowest level of the model and is the communications interface between model and Deep Web sites. Access Schedule is responsible for selecting candidate websites with the highest relevance, while Web Crawler gives the parallel and real-time visit to the websites and obtains initial pages.

Extraction Layer: It is responsible to analyze the structural and logical characteristics of HTML pages,

0-7695-3072-9/07 $25.00 © 2007 IEEE
DOI 10.1109/CIS.2007.39

to identify the meaning of the data unit and to generate raw data sets.

Decision Layer: It charges in cleaning the raw data sets and getting the final data sets, and then sorting the results according to the records of user intendancy.

Application Layer: At the top of the model, this layer is the communication interface between model and users. It is responsible for receiving query requests from users and illustrating the results using the form that users expect, which help users to view or do further processing.

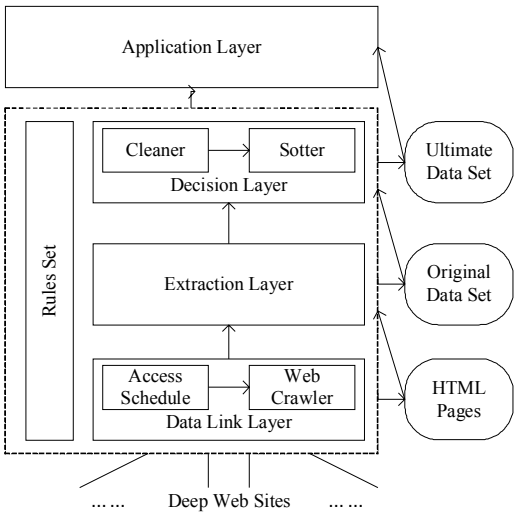

Figure 1. System model

2.2 Rules Set

In this paper, we discuss a set of rules based on Deep Web, which can be divided into three categories:

Structure Rules: When surfing Internet, we usually analyze and understand the web page using the features of some elements, which include the positions and the relationship between them. So we use the following structure rules to help our system to extract data:

1) Some descriptive words lie at the left side or upside of the input controls.

2) Other descriptive words lie inside the input controls. They are usually default values within certain range.

3) The HTML codes of descriptive words adjoin the ones of relevant controls.

4) When a table comprises a whole row of descriptive words that follows with several rows of data, each column of data corresponds to the descriptive cell in the same column.

Logical Rules: They are recognized knowledge, which are used to verify the work in process of information cell location and data extraction.

Application Rules: Since the lowest three layers in the system model are constructed basing on the application layer and provide service to it, the application rules can help to improve the accuracy of the work in the module of Access Schedule, Data Extraction and Data Decision.

The rules discussed above are applied to several major components of the model. The structure and logical rules can be acquired by the method of machine learning and the application rules set are established by the designer of a real system.

Figure 2 shows a general page including several discussed rules.

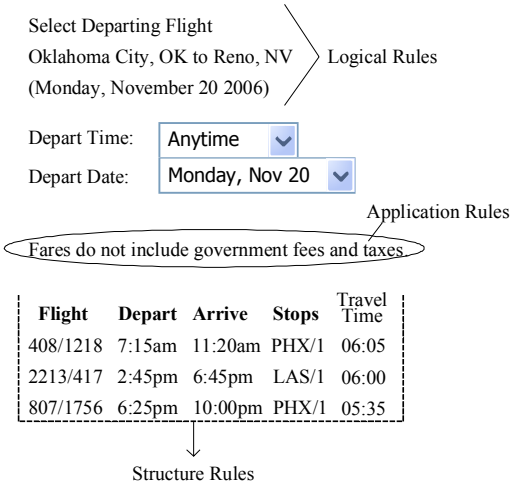

Figure 2. Sample of Deep Web rules

3. Data Link Layer

3.1 Access Schedule

For a query request, not all Deep Web sites candidates contain required results. To predict an effective result set that contains target sites before crawlers depart can significantly reduce the time and space expenditure and promote the efficiency of crawlers.

In this paper, the predication is based on the combination of Application Rules and query interface similarity.

Firstly, we can filter candidate sites using the prior knowledge of a specific field. For example, the specific conditions that users fill in can help to remove some obvious inconsistent sites. Secondly, we compare the similarity between users' query conditions and Deep Web sites that are in the newly acquired candidate pool, and choose the sites with the highest similarity for further query.

For query interfaces, the meaningful information involves the descriptive words as well as the type and range of data source controls. In this paper, the definition of similarity between two interfaces A, B is as follows:

$$S_{AB} = \frac{\sum_{k=1}^{\min\{i,j\}} \omega_{Ak}\omega_{Bk}}{\sum_{k=1}^{\max(i,j)} \omega_k^2} \quad (1)$$

where ω is the weight of a inquiry condition, which can be defined as:

$$\omega_i = Ti + H_i + \sum^{M} FREQ \quad (2)$$

where Ti is the similarity between the descriptive words of Ai and specific knowledge, Hi represents the type of data source controls of Ai, FREQ represents the frequency characteristics of each control value and M is the number of the considered controls.

Let Θ_s be the threshold of query tasks, then

$$\lambda_{si} = \begin{cases} 1 & S_{0i} > \Theta_s \\ 0 & others \end{cases} \quad (3)$$

When λ_{si} = 1, the i-th site that crawlers can access might include valid results, which can be called a valid site. System server visits a set of these valid sites concurrently to improve the efficiency of the system.

3.2 Web Crawler

This paper uses the classic crawler technology to visit Deep Web site and takes account of the following two special situations.

The first special situation is that some web sites set Session Information to track users and put it into the server addresses. However, the string of random digits causes the uncertainty of server address. In this paper, the objective addresses of crawlers are constructed dynamically. We divide the address into two parts----fixed and variable one, where the fixed one records the static address of a site gathered in the phase of the training and the dynamic one comes from practical data that is obtained when crawlers visit a certain site each time.

Another special situation is that in some websites, it is not the result pages containing required information, but the intermediate pages are returned after the forms are submitted. Because of special description information in intermediate pages, we expand the Application Rules to make crawlers do a two-step interactive task: The first step is to detect and acquire the intermediate pages; the second step is to retrieve the useful information from the intermediate pages to form the second query request, and thus to obtain the final result pages.

4. Extraction Layer

According to training results, we find that most of the information is stored in the regions having table-like structures. So we can convert a page to a Web Page Tree, in which the internal nodes represent the structure of the page and the leaf nodes preserve the information. In this section, we propose a novel data extraction algorithm by matching a Web Page Tree with the tree acquired from the configuration file, which can be called Configuration Tree. Moreover, structure rules are used to extract data from HTML pages, and the logical and application rules are applied to correct the extraction results. The data extraction algorithm can be dived by the following five steps:

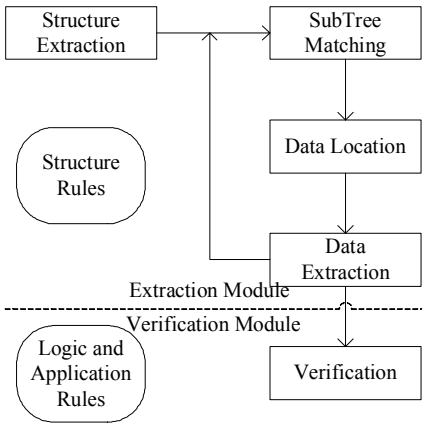

Figure 3. Extraction layer

Extraction Module:

1) From the first leaf node, compare with the numbers of its brother nodes on the two trees. If the numbers are equal, judge whether the attributes of their father nodes are identical with each other. If identical, continue the comparison in a higher level until we find a node that has been defined as a terminal one.

2) Mark the nodes of a sub-tree in the Web Page Tree as the symbol 'Matched', if the sub-tree matches at the first step. Otherwise, mark them as the symbol 'Not Matched'.

3) For each sub-trees marked as 'Matched', repeat 1) and 2) to locate the information cells.

4) Continue the above steps from the next unmarked leaf node until all leaf nodes are compared.

Verification Module:

5) Verify the extracted results using the logical and application rules learned before.

The algorithm is a kind of fuzzy matching algorithm which is flexible and robust, since the data cells can still be located precisely even though there are minor differences between two sub-trees.

5. Decision and Application Layers

After receiving the original data set submitted by extraction layer, data decision layer works based on the Application Rules to optimize the results.

Firstly, data cleaner removes the invalid records which miss some major attributes or are inconsistent with the facts. Secondly, it combines the redundant records, which are caused by the cooperation of different sites. Finally, it unifies the format of the original search results that come from the different sites with various formats.

The sorter is used to evaluate the importance of the attributes of results and sort them basing on the application rules which come from the searching tendency of users. In this paper, a sorting agent is designed to complete this work. It is responsible to collect users' query record and operating log to analyze and predict expected query of users. At the same time, it orders the results according to dynamic weights of domain knowledge and users' tendency.

6. Experiment Result

To evaluate the accuracy of the proposed model, we apply it to three intelligent systems, scientific paper retrieval, electronic ticket ordering and resume searching.

Let Nd be the number of sites that contain the valid results, which are identified by human inspection and Np be the number of sites that Data Link Layer choose to visit accurately, then the rate of adoption is defined as:

$$Cth = Nd / Np \qquad (4)$$

This shows the capacity of Access Schedule.

Let Nf be the number of valid records that are identified by human inspection and Nk be the number of the results that are extracted by the proposed system, then

$$Cac = Nk / Nf \qquad (5)$$

This represents accuracy of Data Extraction.

The experimental results are shown in Table 1:

Table 1. Experiment Result

Test Set	Number of Query Condition	Access Schedule Capability			Data Extraction Capability		
		Nd	Np	Cth	Nf	Nk	Cac
Paper	42	215	215	100%	3928	3847	97.9%
e-Ticket	56	437	420	96.1%	1604	1581	98.6%
Resume	32	246	242	98.4%	1305	1268	97.2%
Total	130	898	877	97.7%	6837	6696	97.9%

7. Conclusions

In this paper, we propose a novel model to extract data from Deep Web pages. The rules of structure, logic and application are applied to improve the classic techniques of data extraction. We apply the model to three hot research areas and the results show that the proposed method is robust and feasible.

We will introduce the method into other web intelligent systems in future work.

References

[1] BrightPlanet.com. The deep web: Surfacing hidden value. Accessible at http://brightplanet.com, July 2000

[2] Chang K.C, He B, Li C, Patel M., Zhang Z. Structured databases on the web: Observations and Implications. *SIGMOD Record*, 2004, 33(3), 61-70

[3] Cope J, Craswell N., Hawking D. Automated discovery of search interfaces on the Web. *In Proceedings of the 14th Australasian Database Conference (ADC 2003)*, Adelaide, 2003, 181-189

[4] Zhang Z., He B., Chang K.C. Under- standing Web query interfaces: best-effort parsing with hidden syntax. *In Proceedings of the 23rd ACM SIGMOD International Conference on Management of Data*, Paris, 2004, 107-118

[5] Arasu A, Garcia-Molina H. Extracting structured data from Web pages. *In Proceedings of the 22nd ACM SIGMOD International Conference on Management of Data,* San Diego, 2003, 337-348

[6] Wittenburg K. Weitzman L. Visual Gram- mars and Incremental Parsing for Interface Languages. *In Proceedings of the IEEE Symposium on Visual Languages (VL)*, Skokie, 1990, 111-118

[7] Liu L, Pu C, Han W. XWRAP: An XML-enabled wrapper construction system for Web information sources. *In Proceedings of the 16th International Conference on Data Engineering*, San Diego, 2000, 611-621

[8] Crescenzi V., Mecca G., Merialdo P. RoadRunner: towards automatic data ex- traction from large Web sites. *In Proceedings of the 27th International Conference on Very Large Data Bases*, Roma, 2001, 109-118

[9] Cohen W. W., Hurst M., Jensen L. S. A flexible learning system for wrapping tables and lists in HTML documents. *In Proceedings of the 11th International World Wide Web Conference*, Budapest, 2002, 232-241

[10] Arlotta L, Crescenzi V, Mecca G, et al. Automatic annotation of data extracted from large Web sites. *In Proceedings of the 6th International Workshop on Web and Databases*, San Diego, 2003, 7-12

[11] Hui Song, Suraj Giri, Fanyuan Ma. Data Extraction and Annotation for Dynamic Web Pages. *In Proceedings of EEE'04*, pages 499~502