A Novel Method for Extracting Information from Web Pages with Multiple Presentation Templates

Li Qingzhong[1]
[1] School of Computer Science and Technology Shandong University, Jinan, P.R. China
lqz@sdu.edu.cn

Ding Yanhui[2], Feng An[3] and Dong Yongquan[2]
[2] School of Computer Science and Technology, Shandong University, Jinan, P.R. China
[3] School of Computer Science and Technology, Shandong Jianzhu University, Jinan, P.R. China
yanhui_ding@yahoo.com.cn, twohorsean@sdjzu.edu.cn, dongyongquan@mail.sdu.edu.cn

***Abstract*—Web information extraction is the key part of web data integration. With the need of e-commerce website and the development of web design, web pages with multiple presentation templates arise. The current web information extraction systems are usually based on single presentation template, so web pages with multiple presentation templates can't be extracted efficiently. This paper focuses on the extraction problem about web pages with multiple presentation templates. Four different kinds of this problem have been considered, and a novel method based on path entropy, presentation regularity and ontology knowledge is presented. The experiment indicates that this method is very promising and it achieves excellent recall and precision.**

***Index Terms*—Information Extraction; Multiple Presentation Templates; Path Entropy; Presentation Regularity; Ontology**

I. INTRODUCTION

The amount of information available on the Internet is tremendous and continuing to grow rapidly. Information represented by Web pages is usually formatted to be easily read by people. The previous web pages were usually organized by single presentation template, which means that all the data records share the same appearance. Recently, in order to meet user's need, web pages organized by multiple presentation templates arise. Data records with the same presentation template share the same look and feel.

Web information extraction has been studied actively to get information efficiently from enormous quantities of web data. Most of the previous works on information extraction [1,6,7,8,9,10,11] only focus on web pages with single presentation template. Web pages with multiple presentation templates can't be extracted efficiently.

In this paper, we present a novel scheme for extracting information from web pages with multiple presentation templates. The key idea is to identify the data records group with the same presentation template, and then construct the wrapper to extract the information from web pages. This paper makes the following contributions. Firstly, we formalize the extraction question on web pages with multiple presentation templates and make a comprehensive analysis of this problem. Secondly, we propose a novel scheme by the organic combination the following heuristics: path entropy, structural and visual layout features of the page. Thirdly, the comparison to the state-of-the-art system clearly shows that our method performs better in most of the cases.

The rest of the paper is organized as follows. Section 2 defines the extraction question on web pages with multiple presentation templates. Section 3 presents a detailed description of our proposed solution. An evaluation of the system as well as a comparison to one of the most important state-of-the-art systems is described in Section 4. Section 5 discusses two special cases of this problem. Section 6 reviews related works. Section 7 concludes the paper.

II. INFORMATION EXTRACTION ON WEB PAGES WITH MULTIPLE PRESENTATION TEMPLATES

Definition 1 (Presentation Template) A presentation template is a collection of factors that will affect the appearance of one data record. Data records with the same presentation template share the same look and feel.

A web page with multiple presentation templates means that more than one presentation template are adopted to display data records in this web page. Different presentation template has different presentation regularities. Data records with the same presentation template share the same look and feel. As showed in Figure 1, the data records located in region 1 share the same presentation template, $T1$. And the data records located in region 2 share another presentation template, $T2$.

Regarding the extraction question on web pages with multiple presentation templates, there are mainly two aspects:

(1) Identify the group in which the data records share the same presentation template.

(2) For different groups, construct the corresponding wrappers to extract web information.

doi:10.4304/jsw.5.5.506-513

For question (2), there have been a lot of research results [2,3,7,11]. Therefore, this paper will focus on the first issue.

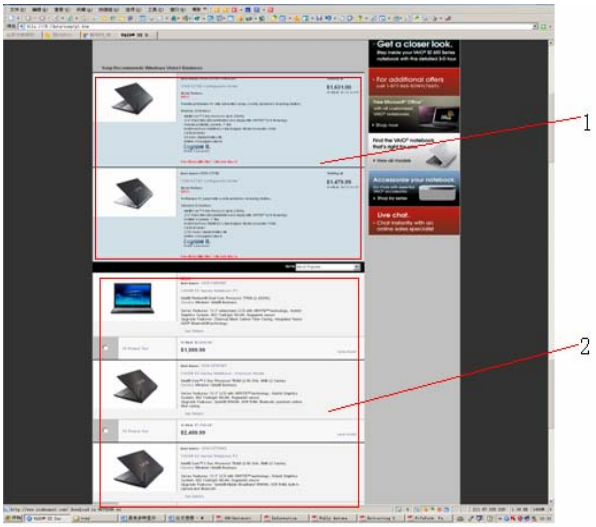

Figure 1. Web pages with multiple presentation templates

Four cases about this question will be discussed in this paper.

Case 1: Data records with the same template are located in the same subtree, t, and there are no other data records with different template in t. An example is showed in Figure 2 .

Case 2: Data records with the same template are located in the same subtree, t, while there are other data records with different template in t. An example is showed in Figure 3.

Case 3: One data record spans several subtrees. These data records are located in the same subtree, t,and there are no other data records with different template in t. An example is showed in Figure 5.

Case 4: One data record spans several subtrees. These data records are located in the same subtree, t, while there are other data records with different template in t. An example is showed in Figure 6.

In order to make a clear structure of this paper, we first concentrate on two common cases, Case 1 and Case 2, in this section. The other two special cases, Case 3 and Case 4, will be discussed in section 5. Figure 2 and 3 show the data organizations of the two common cases.

For notational convenience, we do not use an actual HTML tag names but ID numbers to denote tag nodes in a tag tree. In Figure 2, nodes 4 and 5 share the same presentation template, while nodes 6, 7 and 8 share another presentation template. In Figure 3, nodes 2 and 3 share the same presentation template, while nodes 4, 5 and 6 share another presentation template.

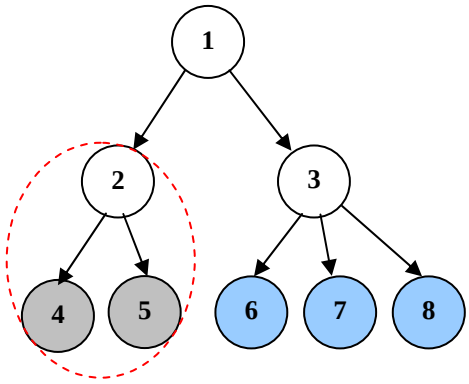

Figure 2. Case 1: One subtree with single presentation template

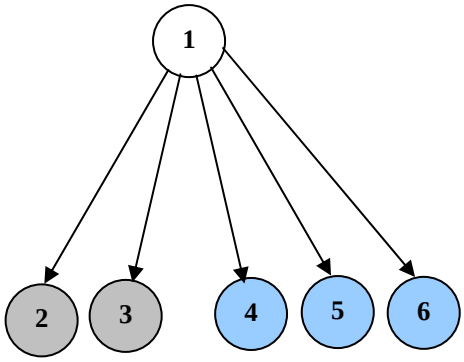

Figure 3. Case 2: One subtree with multiple presentation templates

III. THE ALGORITHM OF MTPE (MULTIPLE PRESENTATION TEMPLATES PAGE EXTRACTION)

The key idea of MTPE is to identify groups in which data records share the same presentation template. Figure 4 shows the architecture of MTPE. A user or an application may submit an URL to MTPE to initiate the information extraction process. The result returned by MTPE is a list of data record groups extracted from the given web page, and data records belonging to the same group have the same presentation template.

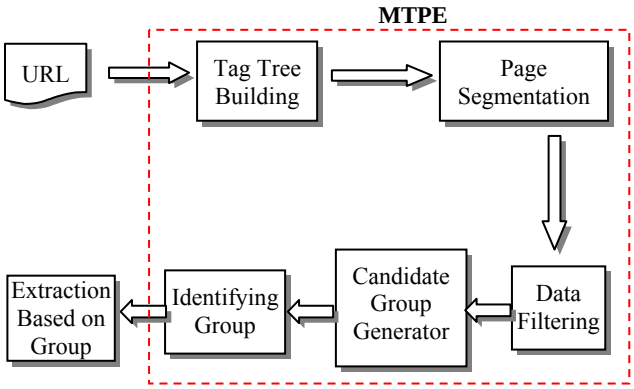

Figure 4. Architecture of our approach

MTPE is based on the following observations:

Observation 1 A group of data records sharing the same presentation template have the similar subtree structure in tag tree.

Observation 2 A group of data records sharing the same presentation template are reflected in the tag tree by the fact that they are under one parent node.

Observation 3 A group of data records sharing the same presentation template locate in the same level in the tag tree.

Observation 4 a well organized or homogeneous system will have low entropy.

We define the notion of entropy of a node in a Tag tree[20] in terms of the uncertainty in the root-to-leaf paths under the node. Our method is based on the observation that a well organized or homogeneous system will have low entropy[20]. We use this principle while traversing the Tag tree from the root of primary content region to leaf nodes in a breadth-first fashion, and split the nodes until each and every segment in the Tag tree is homogeneous.

Definition 2 (Path Entropy) The path entropy $\mathrm{Hp(N)}$ of a node N in tag tree can be defined as

$$Hp(N) = -\sum_{i}^{k} p(i)\log p(i)$$

Where $p(i)$ is the probability of path Pi appearing under the node N.

MTPE consists of the following five steps:

Step1: Parse the web page into a tag tree;

Step2: Construct the Visual Block tree and segment the page into several sections, then locate the primary content region;

Step3: Filter the primary content region, and then calculate the path entropy of each internal node;

Step4: Get the candidate group in which the data records share the same presentation template;

Step5: Identify the group with same presentation template.

A. Tag Tree Building

This phase is dedicated to preparing the web document for extraction. It takes an URL from an end-user or an application, and performs the following three tasks: First, the web page specified by the URL will be fetched from a remote site. Second, the fetched page will be cleaned using the syntactic normalization algorithm, which transforms the given web page into a well-formed web document. Third, the well-formed web document will be converted into a tag tree representation based on the nested structure of start and end tags.

B. Page Segmentation

In this phase, we will locate the minimal subtree[2] that contains all the data records of interest in a page. Web pages are designed for human browsing. So in addition to the primary content region, many web pages often contain other information such as advertisements, navigation links, and so on. Therefore, given a web page p, the first task is to identify which part of p is the primary content region[2]. Let t be the tag tree of Web page p, then the task of locating the primary content region is reduced to the problem of locating the subtree of t which contains all the data records of interest. Obviously, there may be more than one subtree that contains all the data records of interest. The main goal of the primary content region discovery is to locate the minimal subtree of t which contains all the data records of interest.

The primary content region usually has the following features[22]. First, it is always located in the middle section horizontally on Web pages. Second, the size of it is usually large when there are enough data records in the primary content region. In our implementation we first adopt the VIPS algorithm[21] to build a Visual Block tree for each Web page. Then the primary content region can be located by finding the block that satisfies the two above features. Each feature can be considered as a rule or a requirement[22]. The first rule can be applied directly, while the second rule can be represented by $(area_b / area_p) \geq \delta$,where $area_b$ is the area of block b, $area_p$ is the area of the current Web page, and δ is the threshold used to judge whether block b is sufficiently large relative to $area_p$ .The threshold δ is trained from sample pages collected from different real web sites. For the blocks that satisfy both rules, we select the block at the lowest level in the Visual Block tree.

C. Data Filtering

After the primary content region is located, data filtering process mainly deletes the useless nodes in the primary content region. Useless node is the node which has nothing to do with the data records, such as separator tag, script tag, annotation tag, etc. In our experiment, this process will be done by *HtmlCleaner*, an open source tool.

After data filtering, the path entropy of each node included in the primary content region can be calculated.

D. Candidate Group Generator

According to observation 2 and 3, FCG (Finding Candidate Group) algorithm traverses the primary content region from the root downward in a breadth-first fashion, and compares the path entropy of sibling nodes, which have the same parent and locate at the same level in tag tree. In our experiment, a threshold , γ , will be defined to judge whether two path entropy has similar value. The threshold is trained from sample Web pages from different real Web sites. Nodes with similar path entropy will be grouped .

Algorithm 1 Finding Candidate Group (FCG algorithm)

Input: Root of the primary content region, denoted as R;

Output: Candidate group.

Begin
01 For all children of R do
02 If ($IsSameGroup(C[i], C[j])$) then
// C is an array, which stores the children nodes of R;
03 Put $C[i], C[j]$ into one candidate group;
04 If there are some children of R, which have never been grouped then
05 Store these nodes into another array, $ungroup$
06 For all the nodes saved in $ungroup$ do
07 FCG($ungroup[i]$);
08 Else
09 Return all the candidate groups;
End

Algorithm 2 IsSameGroup

Input: Sibling nodes i and j, the threshold of path entropy similarity, γ;

Output: Whether two nodes belong to the same candidate group.

Begin
01 if |PahtEntropy[i]-PathEntropy[j]|< γ then
//PathEntropy[k] stands for the path entropy of node k
02 return true;
03 else
04 return false.
End

E. Identifying the Group

The data records sharing the same presentation template may have similar path entropy, but some other subtree including other presentation information may also have the similar path entropy. So two data records can't be ascertain whether they share the same presentation template only according to the similarity of their path entropy. Some refinement step is needed to ascertain the data records with same presentation template.

The key idea of refinement process is to ascertain the group by taking advantage of presentation regularity included in the subtree of each data records. Presentation regularity is a sequence of the presentation tags[20] by preorder traversal on the corresponding subtree of each data records. For example, a root-to-leaf path *html.body.table.td.b.a* of a label l reveals that l is bold and has a link. The presentation regularity is b-a.

According to the observation (1), data records sharing the same presentation template have similar subtree structure. Since presentation regularity is a part of subtree structure, so if the structures of two subtrees are similar, then the presentation regularities of two subtrees are also similar.

Algorithm 3 Identify Group

Input: Node i, j and the threshold μ. (The two nodes have similar path entropy)

Output: whether two nodes really belong to the same group.

Begin
01 Generate the presentation regularity of each subtree, denoted as d_i, d_j;
02 Compute the edit distance between d_i and d_j, denoted as Distance;
03 if (Distance< μ) then
04 return true;
05 else
06 return false.
End

Ⅳ. EXPERIMENTS

This section we will evaluate MTPE with other Web information extraction system. In order to facilitate comparison with other methods, in our experiment we first ascertain the data records group with the same presentation template by MTPE, and then we construct the wrapper by Roadrunner[1], an open source tool, to extract each data records of each group.

Since most of the current web information extraction systems have not focused on the page with multiple presentation template , so they performs badly. In this paper we only compare MTPE with the famous MDR system[3]. For web pages with single presentation template, MTPE performs as well as MDR. For web pages with multiple presentation templates, MTPE also performs very well. Our experimental results are given in Table 1.

Column 1 lists the site of each test page. It consists of 10 web pages with single presentation template and 6 web pages with multiple presentation templates. We did not collect more web pages with multiple presentation templates for testing because such pages are relatively rear and quite difficult to find. Column 2 gives the numbers of correct data records. Column 3 and 4 give the numbers of correct data records extracted by MDR and MTPE from each page respectively. Column 5 gives the numbers of data records (including correct data records and incorrect data records) extracted by MTPE. Column 6 gives the remark. Err. represents the incorrect data records extracted by MTPE. Miss. represents the data record which is not extracted.

We use the recall and precision measures, which are widely used to evaluate information retrieval systems, to evaluate the performance of our system for extracting data records. Define A as the total number of data records to be extracted, B as the number of data records which

are correctly extracted, C as the number of data records which are wrongly extracted.

$$\mathrm{Re}call = \frac{B}{A},$$

$$\mathrm{Pr}ecision = \frac{B}{B+C}$$

The result shows that our system can achieve high extraction accuracy in both two types of pages. On the other hand, the shortcomings of MTPE include, (1) MTPE requires that web page must contain at least two data records with same presentation template. Because MTPE identifies data records by comparing subtree structure and appearance similarity. If only one data record exists, MTPE can't identify it efficiently. (2) The thresholds of path entropy similarity and presentation regularity similarity play an important role in extraction results. MTPE can't predefine exactly the two thresholds, which means that for different web site a set of training pages is needed to decide the two thresholds.

Since new features arise in some websites , such as the website of sonystyle, MDR can not extract any correct data records and the error rate of MTPE is also very high. The main reason is that one single data record spans several subtrees (showed in Figure 5 and 6). This problem will be discussed in next section and solved by a new heuristic.

TABLE I. EXPERIMENT RESULTS (1)

URL	Data Records	MDR	MTPE		
			Corr.	Found	Remark
Web pages with single presentation template					
http://www.barnesandnoble.com	10	9	10	11	1 err
http://www.tigerdirect.com	18	18	16	16	2 miss
http://www.bookpool.com	10	10	10	10	
http://review.zdnet.com	6	6	6	6	
http://ieeexplore.ieee.org	25	25	25	25	
http://www.shopping.com	30	30	30	30	
http://www.google.com/products	10	10	10	11	1 err
http://www.target.com	12	11	12	13	1 err
http://www.compusa.com	18	18	16	16	2 miss
http://shop.lycos.com	20	20	20	20	
Total	159	157	155	160	
Recall/Precision		98.7% / 100%	97.4% / 96.8%		
Web pages with multiple presentation templates					
http://www.sonystyle.com	7	0	2	14	12 err
http://www.drugstore.com	6	4	6	6	
http://www.newegg.com	13	12	13	14	1 err
http://www.travelocity.com	18	4	18	20	2 err
http://www.kidsfootlocker.com	7	4	6	6	1 miss
http://www.kmart.com	24	4	24	24	
Total	75	28	69	84	
Recall/Precision		37.3% / 100%	92% / 82.1%		

Ⅴ. TWO SPECIAL CASES

The experiment shows that MTPE works badly at the following two cases (Case 3 and Case 4). The main reason is that in these Web pages one data record spans more than one subtree, and only with the path entropy MTPE can not judge the data records sharing the same presentation template correctly .So that the extraction process fails. The two cases (Case 3 and Case 4, which are introduced in Section 2) are showed in Figure 6 and Figure7. Each red dashed region stands for one data record, which is composed of two subtrees.

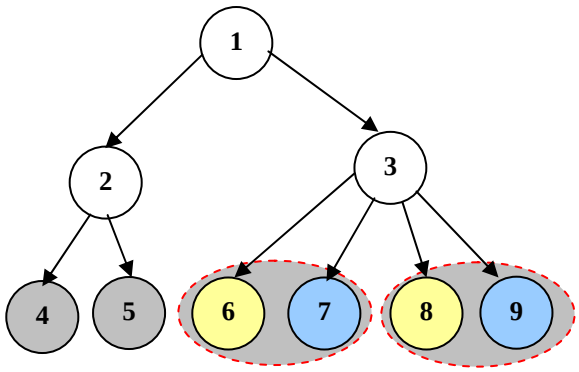

Figure 5. Case 3: Data record spanning more than one subtree

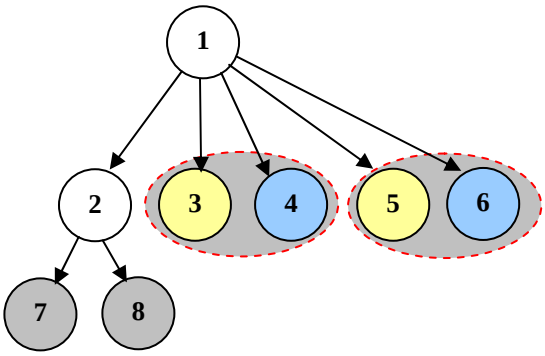

Figure 6. Case 4: Data record spanning more than one subtree

An ontology-matching heuristic[4] is proposed here to identify each data record. The description of this method, named as OntologyAssistant, is showed as follows:

Step1: Build the domain ontology and generate record identifying fields.

Step2: Evaluate the number of data records by multiple record identifying fields.

Step3: Identify the data record separator tags and separate each data record.

Step4: Identify the group, and then construct wrappers to extract data records.

We may expect one or more fields of a record to appear once and only once in the record. We call such fields record-identifying fields[4].We note that a record-identifying field is not the same as a key for a record, but rather is a field that is likely to occur once and only once for each record.

For each record-identifying field, if we can locate a value for the field or even just an indication that the value exists, we can count the number of such occurrences. Then, if we take the average number of occurrences for several record-identifying fields in a Web page p , we have a good chance of correctly estimating the number of records in p . With this estimate, we can rank the candidate separators by how closely their number of appearances corresponds to the estimated number of records. Then the data records can be separated and the groups can be identified by presentation regularity. Finally the extraction task can be fulfilled by existing methods.

As a result of page length limit, the detail is omitted. The experiment result is showed in Table 2. With the aid of OntologyAssistant algorithm, MTPE can solve effectively the cases of single data record spanning several subtrees.

TABLE II. EXPERIMENT RESULTS (2)

	MTPE	MTPE+OntologyAssistant
Recall	92%	98.6%
Precison	82.1%	96.1%

Ⅵ. RELATED WORKS

There have been a lot of researches on web information extraction recently. Readers may refer to [5,19] for a survey on major earlier works. Works on wrappers[14,15,16] provide learning mechanisms to mine the extraction rules of documents. The WebKB project in [17] automatically creates a computer understandable knowledge base from the textual content and hyperlink connectivity of Web pages. Works in [13, 18] provide auxiliary systems to aid in the information extraction from semi-structured documents .

The models presented by IEPAD[6], RoadRunner [1]，EXALG[7]，DELA[8]，PickUp[9]，Viper[10] and DEPTA[11] did not pay attention to the web pages with multiple presentation templates, which is the main focus of this pager, so that they can not extract such web pages efficiently.

MDR[3] has the ability to extract web pages with multiple presentation templates, but MDR works well only for table and form enwrapped records while our method does not have this limitation. MDR proposes a technique based on two observations about data records on the Web and the use of a string matching algorithm. The first observation is that a group of data records containing descriptions of a set of similar objects are typically presented in a particular region of a page and

are formatted using similar HTML tags. The HTML tags of a page are regarded as a string; therefore, a string matching algorithm is used to find those similar HTML tags. The second observation is that a group of similar data records being placed in a specific region is reflected in the tag tree by the fact that they are under one parent node which must be found. The proposed technique is able to mine both contiguous and noncontiguous data records in a Web page.

Another similar system is Vints[12]. Vints proposes an algorithm to find SRRs (search result records) from returned pages of the search engines. However, our method focuses on list pages with multiple presentation templates. Vints treats a web page as a collection of content lines and identifies those content lines that separate the web page into several blocks. Then blocks are grouped by their visual similarities for subsequence analysis. Vints only works best on search engine results.

Ⅶ. Conclusions and Future Work

In this paper, we conduct an in-depth analysis on web pages with multiple presentation templates. And a novel schema based on path entropy and presentation regularity is put forward. Our method first parses the web page into a tag tree and calculates the path entropy of internal nodes; second segments the web page into several parts and identifies the primary content region; third finds the candidate group in which data records share the same presentation templates; finally, identifies the group. For the cases that one single data record spans several subtrees, an ontology-matching heuristic is proposed to solve them successfully.

In the future we intend to focus on the following two questions:

(1) How to identify one single data record with some presentation template effectively;

(2) How to build ontology automatically by ontology learning and further to improve the automated level of our method.

Acknowledgment

The research was supported in part by the National Natural Science Foundation of China under Grant No.60673130; and by National Key Technologies R&D Program of Shandong Province under Grant No.2005GG3201088; and by Science and Technology Planning Project of Shandong Province of China under Grant No.2006GG2201052.

References

[1] Crescenzi V, Mecca G, Meraldo P. RoadRunner: Towards automatic data extraction from large Web sites. In: Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB 2001). Roma: Morgan Kaufmann Publishers, 2001. 109−118.

[2] Buttler, D., Liu, L., Pu, C. A fully automated extraction system for the World Wide Web. IEEE ICDCS-21, 2001.

[3] Liu B, Grossman R, Zhai Y-H. Mining Data Records in Web Pages. SIGKDD'03, 2003.

[4] D.W. Embley, Y. Jiang, and Y.K. Ng, Record-Boundary Discovery in Web Documents, Proc. 1999 ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), 1999.

[5] Chang C-H, Kayed M, Girgis M-R, Shaalan K. A survey of Web information extraction systems. IEEE Trans. on Knowledge and Data Engineering, 2006,18(10):1411−1428.

[6] Chang C, Lui S. IEPAD: Information Extraction based on Pattern Discovery. World Wide Web Conference, 2001.

[7] Arasu A, Hector G M. Extracting structured data from Web pages. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. San Diego: ACM Press, 2003. 337−348.

[8] Wang J Y, Lochovsky F H. Data extraction and label assignment for Web databases. In: Proc. of the 12th Int'l World Wide Web Conf. (WWW 2003). Budapest: ACM Press, 2003. 187−196.

[9] Chen L, Jamil H, Wang N. Automatic Composite Wrapper Generation for Semi-Structured Biological Data Based on Table Structure Identification. SIGMOD Record, June 2004.

[10] Simon K, Lausen G. ViPER: Augmenting automatic information extraction with visual perceptions. In: Proc. of the ACM CIKM Int'l Conf. on Information and Knowledge Management. Bremen: ACM Press, 2005. 381-388.

[11] Zhai Y-H, Liu B. Structured data extraction from the Web based on partial tree alignment. IEEE Trans. on Knowledge and Data Engineering, 2006,18(12):1614−1628.

[12] Zhao H-K, Meng W-Y, Wu Z-H, Raghavan V, Yu C. Fully automatic wrapper generation for search engines. In: Proc. of the 14th Int'l Conf. on World Wide Web (WWW 2005). Chiba: ACM Press, 2005. 66−75.

[13] B. Adelberg, NoDoSE—A Tool for Semi-Automatically Extracting Structured and Semistructured Data from Text Documents, Proc. 1998 ACM SIGMOD Int'l Conf. Management of Data(SIGMOD), 1998.

[14] W. Cohen, Recognizing Structure in Web Pages Using Similarity Queries, Proc. Nat'l Conf. Artificial Intelligence(AAAI), 1999.

[15] N. Kushmerick, D. Weld, and R. Doorenbos, "Wrapper Induction for Information Extraction," Proc. 15th Int'l Joint Conf. Artificial Intelligence (IJCAI), 1997.

[16] W.Y. Lin and W. Lam, Learning to Extract Hierarchical Information from Semi-Structured Documents, Proc. ACM Ninth Int'l Conf. Information and Knowledge Management (CIKM), 2000.

[17] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, Learning to Construct Knowledge Bases from the World Wide Web, Artificial Intelligence, vol. 118, nos. 1-2, pp. 69-113, 2000.

[18] C.N. Hsu and M.T. Dung, Generating Finite-State Transducers for Semi-Structured Data Extraction from the Web, Information Systems, vol. 23, no. 8, pp. 521-538, 1998.

[19] A. Laender, B. Ribeiro-Neto, A. Silva, and J. Teixeira, A Brief Survey of Web Data Extraction Tools, SIGMOD Record, vol. 31, no. 2, June 2002.

[20] Vadrevu, S., Gelgi, F. and Davulcu, H. Information extraction from web pages using presentation regularities and domain knowledge. In: Proc. of the 16th Int'l Conf. on World Wide Web(WWW 2007). Banff, Alberta, CANADA. pp.157-179.
[21] Cai, D., Yu, S., Wen, J. and Ma, W. VIPS: a vision based page segmentation algorithm. Technical report, Microsoft Technical Report, MSR-TR-2003-79 (2003).
[22] Liu, W., Meng, X., Meng, W. Vision-based Web Data Records Extraction. In: Proceedings of the 9th SIGMOD International Workshop on Web and Databases (SIGMOD-WebDB 2006), Chicago, Illinois, June 30 (2006)

Li Qingzhong is Professor in the School of Computer Science and Technology at Shandong University. His primary research includes: Information integration and SAAS.

Professor Li is an advanced member of Chinese Computer Federation.

Ding Yanhui is a Ph. D. Candidate. He studies at the School of Computer Science and Technology, Shandong University. His primary research is Web information integration.

Feng An is a lecturer in the School of Computer Science and Technology at Shandong Jianzhu University. His primary research is digital image processing and data integration.

Dong Yongquan is a Ph. D. Candidate. He studies at the School of Computer Science and Technology, Shandong University. His primary research is Web information integration.