A Personal Web Information/Knowledge Retrieval System

Hao Han and Takehiro Tokuda
{han, tokuda}@tt.cs.titech.ac.jp

Department of Computer Science, Tokyo Institute of Technology
Meguro, Tokyo 152-8552, Japan

Abstract. The Web is the richest source of information and knowledge. Unfortunately the current structure of Web pages makes it difficult for users to retrieve the information or knowledge in a systematic way. In this paper, using the tree approach, we propose a personal Web information/knowledge retrieval system for the extraction of structured parts from Web pages. First we get the layout pattern and paths of extraction parts of a typical Web page in target sites. Then we use the recorded layout pattern and paths to extract the structured parts from the rest of Web pages in target sites. We show the usefulness of our approach using the results of extracting structured parts of notable Web pages.

1 Introduction

Today Web information/knowledge retrieval by a personal user is usually done through the use of Web browsers with the help of search engines’ index information. However, if we would like to get information and knowledge from a collection of necessary partial information of Web pages in one Web site or a number of Web sites, the use of Web browsers may not be a good solution.

For example, in the BBC country profiles site, there exists a collection of 200 or more country/region information including most recent basic information such as capital city, population and leader’s name. If we would like to retrieve a collection of necessary basic information of 200 or more countries/regions, the use of Web browsers would be a time-consuming tedious task. Similar personal information/knowledge retrieval tasks may be retrieval of a collection of disease names and corresponding parts of human body from health/medicine sites or retrieval of a collection of company names and corresponding industrial area from finance sites.

The purpose of this paper is to present a system for personal Web information/knowledge retrieval. Our system allows users to automatically collect necessary partial information or whole information from Web pages in one or a number of Web sites. What users have to specify may be the starting Web page, crawling area, target parts selection for one typical Web page, and the resulting table organization.

The organization of the rest of this paper is as follows. In Section 2 we give an overview of our system. In Section 3 and 4 we respectively explain the method of partial information extraction and the method for reuse of layout patterns and paths. In Section 5 we show two kinds of resulting tables to present the extracted information. In Section 6 we give examples
of Web information/knowledge retrieval using our system. In Section 7 we discuss related work and evaluate our system. Finally we give our concluding remarks in Section 8.

2 Overview

Our personal Web information/knowledge retrieval system has following steps for a user to retrieve a collection of partial information or whole information of Web pages in one Web site or a number of Web sites.

Step 1. Specification of start points and crawling scopes in the target Web sites
Step 2. Definition of names of target parts and their data types
Step 3. Acquisition of layout pattern and selection of partial information or whole information from a typical Web page in the target Web sites
Step 4. Reuse of the selection pattern of partial information in ordinary Web pages of the target Web sites
Step 5. Definition of the resulting table format

The outline of our system is shown in Fig. 1. We use XML tree approach for the extraction of partial information.

![Figure 1: Outline of our system](image)

3 Extraction of Partial Information

3.1 Definition of Part Names and Data Types

We define a name for each target part and its data type for the extraction and presentation of partial information. The data type includes two kinds of information: property and structure. Property is text or object. Text is the character string in Web pages such as an article. Object is one instance of the photo, video and other multimedia files. Structure is single occurrence or continuous occurrence. A single occurrence is a node without similar sibling nodes such as the title of an article, and the continuous occurrence is a list of similar sibling nodes such as the paragraphs of an article. There are four kinds of data types: single text, continuous text, single object and continuous object. For example, for a news article Web page, the news title is a single text with name ”title”, the news contents are continuous text with name ”paragraph” and one photo is a single object with name ”photo”.
3.2 Layout Patterns

3.2.1 Definition of Layout Patterns

The HTML document of a Web page can be represented by a tree structure as shown in Fig. 2. One node can be represented by its path from the root. We define a layout pattern of a Web page for dealing with Web page layout similarity.

A tree structure of a HTML document can be divided into a number of subtrees. A Layout Pattern is a list of paths from the root of the entire tree to the roots of these subtrees.

For example, a Web page can be divided into a number of main parts as shown in Fig. 3. The layout pattern of this Web page is the list of paths from the root of the entire tree to roots of all main parts.

![Figure 2: A tree structure and paths](image1)

![Figure 3: A Web page and its divided parts](image2)

3.2.2 Layout Pattern Acquisition

In order to acquire the layout pattern, we need to parse the tree structure of a given HTML document of a Web page. We use JTidy [2] to transform HTML documents into XML documents because of potential syntax errors such as missing end-tags in HTML documents.

We need to define the default number of division of the entire tree into main parts and also the default method of the division of the entire tree. If the number of divided main parts of a Web page is too large or too small, then the list of paths to roots of these main parts may be too sensitive or too insensitive. We analyzed many typical Web pages and found that the square root of the sum of leaf nodes of the tree structure seems appropriate for our extraction of partial information. Our default method of Web page division is as follows.

```java
1. Node ROOT = root node of tree;
2. int SUM = sum of leaf nodes;
3. int MAX = sqrt(SUM);
4. List nodelist = new List();
5. nodelist.add(ROOT);
6. List L = new List();
7. Node nextnode = null;
8. while (L.size() + nodelist.size() < MAX){
9.     L.add(nodelist.getAllNodes());
10.    L.remove(nextnode);
11.    nodelist = nextnode.getChildNodes();
12.    nextnode = the node in L with the maximum leaf nodes;
13. }
```

The nodes in List L are root nodes of the divided subtrees. Usually the visible information is embedded between the node <body> and </body>, so we can consider the node <body> as the root node of tree structure of HTML document. Therefore, the layout pattern is a list of paths from the node <body> to nodes in List L. The path takes a form: body: 0/n1 : o1/n2 : o2/.../n(N − 1) : o(N − 1)/nN : oN, where, nN is the node name of the N-th node, oN is the order of the N-th node among the sibling nodes, and n(N − 1) is the parent node of nN.
3.3 Parts Selection

We select the target parts to reach the partial information. We collect the paths of the selected parts using the following process.

1. We divide the Web page into parts by our default method during the layout pattern acquisition.

2. We judge whether a target part is one of the divided parts. We redivide the part if this part contains both the target part and other undesired parts until the target part becomes a single part.

3. We select the target parts and save the paths of parts as a form: \( \text{body} : 0 : ID/n : o1 : ID1/n2 : o2 : ID2/.../n(N-1) : o(N-1) : ID(N-1)/nN : oN : IDN \), where, \( nN \) is the node name of the \( N \)-th node, \( oN \) is the order of the \( N \)-th node among the sibling nodes, \( IDN \) is the ID value of the \( N \)-th node, and \( n(N-1) \) is the parent node of \( nN \).

3.4 Partial Information Extraction

3.4.1 Path Selection

We need to select the layout pattern corresponding to the Web page using the following steps:

1. We transform the HTML document to the XML document.

2. We select the saved layout pattern one by one.

3. We find out a layout pattern corresponding to the XML document if all the paths in this layout pattern can be found in the XML document.

Then, we regard that the list of paths corresponding to the found layout pattern is the paths of partial information of this Web page.

3.4.2 Subtree Extraction

We extract the subtrees according to the corresponding paths, and every subtree represents a part of Web page. If the data type of a part is continuous occurrence, the corresponding sibling trees with the same node names and ID are extracted, too.
3.4.3 Text Extraction

According to the defined data types, we extract the partial information from the extracted subtrees in text format excluding the tags of HTML document.

For the single text type, the partial information is the node value of the corresponding single leaf node. For the single object type, the partial information is the attribute value of corresponding single node. For the continuous text type, the partial information is the list of extracted values from the corresponding list of single subtrees. For the continuous object type, the partial information is the list of values extracted from the list of continuous subtrees. For example, the extracted information of a photo is the value of attribute “src” of node `<img>`, and the extracted information of the list of paragraphs of an article is the list of values of continuous leaf nodes.

4 Reuse of Layout Patterns and Paths

4.1 Reuse of Layout Patterns

If we can find the similar paths in a layout pattern of the HTML document of a Web page, this Web page may be similar to the Web page corresponding to this layout pattern and this layout pattern may be reused. We give a definition of similar paths of layout pattern.

**Similar Path of Layout Pattern:** Two paths are similar to each other, if these two paths have the same forms ignoring the difference of orders of nodes among sibling nodes, and the difference of orders is within a defined deviation range. The form of path is as follows:

- `body : 0/n1 : (o1 − H ∼ o1 + H)/n2 : (o2 − H ∼ o2 + H)/.../n(N − 1) : (o(N − 1) − H ∼ o(N − 1) + H)/nN : (oN − H ∼ oN + H)`, where, `nN` is the node name of the N-th node, `oN` is the order of the N-th node among the sibling nodes, `n(N − 1)` is the parent node of `nN`, and `H` is the deviation value.

For example, `body : 0/form : 0/table : 1/tr : 0/td : 0` is similar to `body : 0/form : 0/table : 2/tr : 0/td : 0` as shown in Fig. 5.

4.2 Reuse of Paths

If we find a layout pattern that can be applied to a Web page, the specified paths corresponding to this layout pattern can be reused to extract the partial information. Firstly, we give a definition of similar path of part and get a list of similar paths.

**Similar Path of Part:** Two paths are similar to each other, if these two paths have the same forms ignoring the difference of orders of nodes among sibling nodes, and the difference of orders is within a defined deviation range. The form of path is as follows:

- `body : 0/ID/tn1 : (o1 − H ∼ o1 + H)/n2 : (o2 − H ∼ o2 + H)/ID2/.../n(N − 1) : (o(N − 1) − H ∼ o(N − 1) + H)/ID(N − 1)/nN : (oN − H ∼ oN + H)/IDN`, where, `nN` is the node name of the N-th node, `oN` is the order of the N-th node among the sibling nodes, `n(N − 1)` is the parent node of `nN`, and `H` is the deviation value.

Then, we use the ID value to choose the most appropriate paths with the minimum deviation value from the deviation range, and reuse them to extract the partial information.
5 Resulting Tables

We need a default resulting table to present the result after we extract the partial information. We have two types of resulting tables: horizontal type and vertical type. A horizontal type resulting table has a number of columns equal to the sum of the number of selected parts. A column is identified by the name of selected parts. The first row is the header row to display the column names. We also have vertical type resulting tables. Examples of resulting tables are shown in Fig. 6.

6 Examples

In this section, we will give some examples to show the process of partial information extraction and presentation.

We extract the partial information from the top news pages of Yahoo! News and CNN.com and present the extracted information in a resulting table.

1. We specify the top page of Yahoo! News with the crawling area.

2. We define the names and specify the data types of the target parts: news title part “YahooNewsTitle” of single text type, news contents part “YahooNewsContents” of continuous text type, and photo part “YahooPhoto” of single object type.

3. We acquire the layout pattern of a typical Web page and divide the Web page to select the target parts.
4. We do the same operations as Step 1∼3 for CNN.com.

5. Our system extracts the partial information and presents the extracted information in a resulting table as shown in Fig. 9.

We can extract the partial information from all kinds of the Web pages, such as company names and industrial classifications from Yahoo! Finance [8], and country names and profile information from BBC Country Profiles [1] as shown in Fig. 10.

Figure 9: A resulting table  
Figure 10: Extracted partial information

7 Evaluation

Our method is one of tree-oriented approaches. There have been research groups that focus on the problem of extracting information from Web pages based on the tree-oriented approaches. Crunch [3] is a HTML tag filter to retrieve the contents from the DOM trees of Web pages. However, the users have to spend much time in configuring a desired filter after analyzing the source of HTML documents of the Web pages. Internet Scrapbook [4] is a system which allows users to create a personal page by clipping parts of Web pages by specifying parts of Web pages to be extracted, which can not be applied to similar Web pages. PSO [7] is an approach to extract the parts of Web pages. It keeps the view information of the extracted parts by using the designated paths of tree structures of HTML documents, and users need to find out the paths from the HTML document of Web page by hand. Similarly, ANDES [5] is an XML-based methodology to use the manually created XSLT processors to realize the data extraction. HTML2RSS [6] is a system to automatically generate RSS feeds from HTML documents that consist of time-series items such as blog, BBS, chats and mailing
lists. [9] can automatically identify individual data records and extract data items from them. The extraction ranges of [6, 9] are limited to the Web pages that consist of list of data items with similar data structures or special data structures.

Our system allows users to compose one resulting table from various parts of Web pages in a number of Web sites. Also our system allows users to select target parts without noticing the explicit tree structure of Web pages. Table 1 shows the performance of our system.

<table>
<thead>
<tr>
<th>Web site</th>
<th>Deviation</th>
<th>Total pages number</th>
<th>1 pattern</th>
<th>2 patterns</th>
<th>3 patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yahoo! News</td>
<td>1</td>
<td>76</td>
<td>74</td>
<td>75</td>
<td>76</td>
</tr>
<tr>
<td>CNN.com</td>
<td>1</td>
<td>31</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>BBC Country Profiles</td>
<td>10</td>
<td>267</td>
<td>251</td>
<td>265</td>
<td>267</td>
</tr>
<tr>
<td>Yahoo! Finance</td>
<td>1</td>
<td>215</td>
<td>214</td>
<td>215</td>
<td>/</td>
</tr>
</tbody>
</table>

**Table 1: Number of correctly extracted Web pages with the number of layout patterns**

8 Conclusion

We have presented our personal Web information/knowledge retrieval system based on XML tree approach. Our system allows users to extract information and knowledge from the partial information of Web pages in one Web site or a number of Web sites. We can easily select the target parts of typical Web pages and reuse the extracted paths to reach the partial information of general Web pages having similar structures.

The contents extracted from Web pages may be used for personal data backup or Web site analysis of data for public purposes. The reproduction or republication of extracted contents may not be allowed. It is important for users of the personal Web information/knowledge retrieval system to conform to all copyright rules of contents on the Web. Our future work would be to provide mechanisms of static or dynamic combination of tasks of partial information/knowledge retrieval including a task of retrieving metadata such as RDF.

References