Relational Web Wrapper: A Web Data Extraction Approach

Neeraj Raheja¹, Dr.V.K.Katiyar²
Associate Professor1, Professor & Head2
1, 2 Department of Computer Science and Engineering, M.M.University, Mullana (Ambala), Haryana, India

Abstract: The information over the internet is growing at a rapid rate, so web data extraction systems are required to extract the required information. One such technique is web wrapper, which is a supervised learning approach in which a template (program) is developed by the programmer to extract some specific data. This research paper provides a web wrapper known as Relational Web Wrapper which extracts related information of the webpage. Finally, the performance evaluation of this web wrapper on the basis of time to extract the data and accuracy provided are shown in results. The results show this web wrapper provides efficient results.

Keywords: Web data extraction, web wrapper, relational data.

I. INTRODUCTION

The world-wide web (WWW) has become one of the most widely used information resources and hence growing at a very fast rate i.e. millions of websites are available today. Because of this rapid growth and large information available web data extraction systems are necessary to use. Web wrapper is one of such technique which is a supervised learning approach where a programmer develops a program (template) and applies it to a number of webpage. Web wrapper may solve a particular problem i.e. may extract a particular pattern of data. Web wrapper translates the relevant data embedded in web pages into a relational (or other regular) structure and store into some specific format like xml, excel sheets etc. Wrappers may be developed manually, through wrapper induction or automatic approach. The manual generation of wrappers is difficult, error-prone and specific, hence semi-automatic and automatic wrapper construction systems are preferable [6, 7, 11, 12]. Applications of web wrappers include comparison shopping where information from multiple online shops is extracted to compare prices, and online monitoring of news over multiple websites or bulletin boards, say for items relevant to a particular topic.

In this paper, we describe a new system named Relational web wrapper which is wrapper induction approach i.e. a program is developed. This wrapper extracts the related data of a webpage which may further be used for extending the information of the webpage or user.

II. LITERATURE REVIEW

Kushmerick et al. [13] provides two distinct categories of web wrappers i.e. finite-state and relational learning tools. The extraction rules in finite-state tools are formally equivalent to regular grammars or automata, e.g. WIEN [9], Soft Mealy [10] and STALKER [1], while the extraction rules in relational learning tools are essentially in the form of Prolog-like logic programs, e.g. SRV [8], Crystal [5], Webfoot [14], Rapier and Pinoccchio [15]. Muslea et al. [1] provides a web wrapper known as STALKER which takes a webpage as input and then searches some tokens which have to be skipped and then add some new tokens which refines or replaces the current set. R. Baumgartner et al. [2, 3] provides a web wrapper known as Lixto, which offers an interactive interface that hides most technical details from a webpage. Lixto does not support a verification set, and does not provide a ranking of candidate tuple sets to decrease user effort. Senellart et al. [4] provides a web wrapper to extract information from the hidden web sources. It searches the HTML codes of the related WebPages over the internet.

III. SYSTEM MODEL

Relational web wrapper takes a webpage as input and extracts the related data of webpage like Title, Summary and Keywords. The extracted data like keyword (related data of the webpage) are used for further searching so that more results may be provided. The extracted data is stored in excel sheets. It provides the user an easy approach to search the related data of the webpage.
II. ALGORITHM FOR RELATIONAL WEB WRAPPER

Suppose start is the starting time to start the process of extracting metadata from the webpage and end is ending time when the metadata had been extracted. M consists of the extracted data and meta_extracted_time is the time to extract the meta data.

**Input:** A webpage W.
**Output:** Metadata M from the webpage W.

Begin

**Phase 1:** Extracting the data from the webpage and calculation of time to extract the data

\[
\text{start} = \text{current time} \quad (\text{Time at which the data extraction starts})
\]
\[
\text{a} = \text{W\textunderscore title}
\]
\[
\text{b} = \text{W\textunderscore location\textunderscore href}
\]
\[
\text{c} = \text{W\textunderscore getElementsByTagName\textunderscore (\textquotesingle meta\textquotesingle)}
\]
\[
\text{M} = \text{a}
\]
\[
\text{For (x = 0 To y = c\textunderscore length; x < y; x++)}
\]
\[
\quad \text{if (c[x\textunderscore name\textunderscore toLowerCase()} == \text{"description"})}
\]
\[
\quad \quad \text{description} = \text{c[x]};
\]
\[
\quad \text{endif}
\]
\[
\text{endfor}
\]
\[
\text{M= M+description}
\]
\[
\text{for (var x = 0, y = c\textunderscore length; x < y; x++)}\{
\quad \text{if (c[x\textunderscore name\textunderscore toLowerCase()} == \text{"keywords"})}\{
\quad \quad \text{description} = \text{c[x]};
\quad \quad \text{endif}
\quad \text{endfor}
\]
\[
\text{end = current time} \quad (\text{Time at which data extraction ends})
\]
\[
\text{meta\textunderscore extracted\textunderscore time} = \text{end} - \text{start};
\]

**Phase 2:** Storing data to excel sheets

Suppose x is namespace for excel workbook.

\[
\text{xmlns: x=\text{"urn\textunderscore schemas\textunderscore microsoft\textunderscore com:\textunderscore office\textunderscore excel"}}
\]
\[
\text{uri = 'data\textcolon\textcolon application/vnd.ms\textunderscore excel\textbase64'}
\]
Create the excel worksheet using x as following:

```xml
<x:ExcelWorkbook>
  <x:ExcelWorksheets>
    <x:ExcelWorksheet>
      <x:Name>{worksheet}</x:Name>
      <x:WorksheetOptions><x:DisplayGridlines/></x:WorksheetOptions>
    </x:ExcelWorksheet>
  </x:ExcelWorksheets>
</x:ExcelWorkbook>
```

```xml
function B(s)
Begin
return window.btoa (unescape (encodeURIComponent(s)))
End
```

```xml
function F(s, c)
Begin
return s.replace (/ {\((w+)\)/g, function (m, p) {return c[p]}) }
End
```

```xml
function P(table, name)
Begin
if (!table.nodeType)
  table = W.getElementById(table)
endif
M = {worksheet: name || 'Worksheet', table: table.innerHTML}
window.location.href = uri + B (F (template, M))
End
```

IV. EXPERIMENTAL RESULTS

The relational web wrapper is developed on JavaScript platform. 3 Websites were developed in HTML and JavaScript for experimentation. The results from one of the websites are shown in fig 2, fig 3 and fig 4. Similar results are also available for other websites.

![Fig 2: Webpage with extracted data](image-url)
For this purpose 3 websites named MMIM, MMEC and MMCE each consisting of 10 WebPages was developed.

**Fig 3:** Extracted data and evaluation time

**Fig 4:** Storing the extracted data to excel sheet

**PERFORMANCE EVALUATION OF RELATIONAL WEB WRAPPER**

For this purpose 3 websites named MMIM, MMEC and MMCE each consisting of 10 WebPages was developed.
WEBSITE: MMIM

Table 1: Evaluation Time to extract the related data from website MMIM

<table>
<thead>
<tr>
<th>Name of Webpage</th>
<th>Average Time to Extract the Data (Number of runs)</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Index.html</td>
<td>4.0</td>
<td>4.2</td>
</tr>
<tr>
<td>Life.html</td>
<td>6.5</td>
<td>5.4</td>
</tr>
<tr>
<td>Prog.html</td>
<td>6.0</td>
<td>5.2</td>
</tr>
<tr>
<td>Student.html</td>
<td>5.0</td>
<td>4.4</td>
</tr>
<tr>
<td>Approach.html</td>
<td>4.5</td>
<td>4.2</td>
</tr>
<tr>
<td>Message.html</td>
<td>4.0</td>
<td>4.4</td>
</tr>
<tr>
<td>Facilities.html</td>
<td>6.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Knowledge.html</td>
<td>6.0</td>
<td>5.6</td>
</tr>
</tbody>
</table>

WEBSITE: MMEC

Table 2: Evaluation Time to extract the related data from website MMEC

<table>
<thead>
<tr>
<th>Name of Webpage</th>
<th>Average Time to Extract the Data (Number of runs)</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Index.html</td>
<td>5.0</td>
<td>5.2</td>
</tr>
<tr>
<td>Life.html</td>
<td>7.5</td>
<td>7.4</td>
</tr>
<tr>
<td>Prog.html</td>
<td>7.2</td>
<td>7.2</td>
</tr>
<tr>
<td>Student.html</td>
<td>6.0</td>
<td>6.2</td>
</tr>
<tr>
<td>Approach.html</td>
<td>5.5</td>
<td>5.4</td>
</tr>
<tr>
<td>Message.html</td>
<td>5.2</td>
<td>5.4</td>
</tr>
<tr>
<td>Facilities.html</td>
<td>6.0</td>
<td>6.2</td>
</tr>
<tr>
<td>Knowledge.html</td>
<td>6.4</td>
<td>6.6</td>
</tr>
</tbody>
</table>

WEBSITE: MMCE

Table 3: Evaluation Time to extract the related data from website MMCE

<table>
<thead>
<tr>
<th>Name of Webpage</th>
<th>Average Time to Extract the Data (Number of runs)</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Index.html</td>
<td>6.2</td>
<td>6.2</td>
</tr>
<tr>
<td>Life.html</td>
<td>5.5</td>
<td>5.4</td>
</tr>
<tr>
<td>Prog.html</td>
<td>6.2</td>
<td>6.2</td>
</tr>
<tr>
<td>Student.html</td>
<td>6.1</td>
<td>6.2</td>
</tr>
<tr>
<td>Approach.html</td>
<td>6.5</td>
<td>6.4</td>
</tr>
<tr>
<td>Message.html</td>
<td>6.2</td>
<td>6.2</td>
</tr>
<tr>
<td>Facilities.html</td>
<td>7.2</td>
<td>6.8</td>
</tr>
<tr>
<td>Knowledge.html</td>
<td>6.8</td>
<td>6.6</td>
</tr>
</tbody>
</table>

Accuracy: Accuracy is measured on the basis of following formula
Accuracy = Data extracted by the relational web wrapper / Data extracted on manual basis (Actual existence)
Table 4: Accuracy measurement to extract the related data from website MMEC

<table>
<thead>
<tr>
<th>Name of Webpage</th>
<th>% age of data extracted by relational web wrapper</th>
<th>% age of data extracted on manual basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index.html</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Life.html</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Prog.html</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Student.html</td>
<td>95%</td>
<td>100%</td>
</tr>
<tr>
<td>Approach.html</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Message.html</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Facilities.html</td>
<td>95%</td>
<td>100%</td>
</tr>
<tr>
<td>Knowledge.html</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

The relational web wrapper provides 100% accuracy in extraction of title and description of document and more than 0.95 accuracy in extracting the keywords of most of the webpages.

V. CONCLUSION AND FUTURE WORK

Relational web wrapper provides an example of wrapper induction system which may extract related data or meta-data of the webpage efficiently and with minimal user effort. The web wrapper extracts data efficiently in terms of time taken to extract the data and accuracy provided. As per the definition of web wrapper the relational web wrapper can be applied only to extract meta-data from the webpage (single work). The work may further be extended to more dynamic content and saving the results to xml format. Accuracy of the web wrapper may be extended by applying some more accurate techniques on the same web wrapper. In future work the relational web wrapper developed may be compared with already existing techniques of web data extraction.

References