Mahesh A. Sale, Pramila M. Chawan, Prithviraj M. Chauhan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp. 313-318

Information Extraction from Web Tables

Mahesh A. Sale*, Pramila M. Chawan, Prithviraj M. Chauhan*****

*(M.Tech. Computer, Department of Computer Technology, Veermata Jijabai Technological Institute, Mumbai-19, Maharashtra, India)

** (Associate Professor, Department of Computer Technology, Veermata Jijabai Technological Institute, Mumbai-19, Maharashtra, India)

***(Project Manager, Morning Star India Pvt. Ltd., Navi Mumbai, Maharashtra, India)

ABSTRACT

Tables in Web pages have become an important resource for knowledge and information. Therefore it is very necessary to develop a system to extract the information from tables in HTML web pages. The system should be able to distinguish between the attribute and their values in the table and transform the information to a form that a computer can understand. In this paper, we focused on extracting tables from large-scale HTML texts. We analyzed the structural aspects of a web table within which we devised the rules to process and extract attribute-value pairs from the table. In the flow of table extraction, Web page Processing, Table Validation, Table Normalization, Table Interpretation and Attribute-value pair formation are discussed. Heuristic rules are employed to identify the tables. We also propose a table interpretation algorithm which captures the attribute-value relationship among table cells.

Keywords - **Attribute-Value pairs, Genuine Tables, Information Extraction, Web Pages**

I. INTRODUCTION

Tabular representation is an important method of presenting information which is most widely used on the web. Table makes the information more readable and understandable to human beings. The process of automation of information extraction from web tables has applications in information retrieval, web mining, summarization and knowledge acquisition. A table is consists of number of rows. Rows are made up of cells. Each cell is either a label cell or a data cell. Label cell refers to attribute name and data cell refers to attribute value. The process of information extraction from web table involves differentiating the label cells from data cells and identifying the associations between them.

Information extraction from web tables refers to the information mining from web pages. This information is to be extracted from a table which is located within a body of HTML text or sometimes plain text. Many schemes of web table mining have been suggested, covering knowledge management to content delivery to mobile devices. The main difference between tables in plain text and the web tables is that the visual interpretation of web tables depends on the markups embedded inside the free text.

The main objective of this paper is to present a comprehensive framework in which a web table is systematically analyzed. Under this framework, a data retrieval scheme is developed which contains some strategies to identify how an attribute is associated with a value.

The rest of this paper is organized as follows. In section 2 we discuss the related work done in this area. In section 3 we explain the basic concepts and terminologies associated with the information extraction from web tables. Section 4 presents an analysis of web tables which guides us to devise the heuristic rules for table interpretation and information extraction. The flow of processes for table extraction is shown in section 5. We concluded the paper in section 6.

II. RELATED WORK

The task of table detection has been done in the past by Chen et al., 2000, Hu et al., 2000, Wang and Hu, 2002. Table detection has been performed by separating the tables that contain relational and logical information from the tables used for formatting the layout. The documents which contain both the real tables and the tables used for layout formatting are input to the system. The output classifies the data into real tables and non-real tables. Table detection task needs to be done before table extraction because it is very important

to extract the information from valid and real table. However the detection logic can be embedded in the extraction step itself.

The RoadRunner System [6, 7] automates the data extraction from Web sites on the basic of similarities in page layout. Based on PAT trees, Chang & Lui [8] proposed an algorithm which detects repeated HTML tag sequences that represents rows of Web tables.

The tables from which the data is to be extracted can be either the plain text ASCII tables or the HTML tables. Extraction of data from ASCII tables is more challenging than that from HTML tables. This is because; in case of plain text ASCII tables we need to interpret the structural information from spaces, tabs and ASCII character sequences. While in case of HTML tables the rows and cells are organized with standard structure of <tr> and <td> tags respectively. Pyreddy and Croft, Hurst and Douglas, Hurst and Nasukawa described the methods for table extraction from plain text. The features that are used to extract information from HTML tables is different and the features like tabs and spaces need not be used to identify the tables.

III. BASIC CONCEPTS AND TERMINOLOGY

This section explain basic concepts about our web table mining framework, and the associated terminology. Terms such as attribute-value pairs, attribute (value) as a collection of labels (quantities), and visual similarity between pieces of text and cells in the table, are all explained there.

3.1 Tables in HTML

A table in HTML begins with a caption which is optional, followed by one or more rows. Each row is made up of one or more cells. The cells in a row are either header cells or data cells. The cells can span one or more rows or columns.

The following tags are used in table:

a) <table…> </table>

b) <tr…> </tr>

c) <td…> </td>

d) <th…> </th>

e) <caption…> </caption>

In our case not all tables are interesting or genuine tables. Genuine tables are those tables which is a two-dimensional grid semantically conveying the logical relationship among the cells [1]. In [1], from the samples of 11,477 tables, only some 15% of them are considered genuine tables.

Table rendering depends on HTML tags within the text structurally and visually appeared in the table. The structure of a table refers to the rows and columns in the table, and mainly how different texts are stored within the cells of the table.

Among this structural aspect the main part that is interesting to us is the part which is having some semantic implications, e.g. rows and/or columns spanning of a cell. On the other hand, tags about visual appearance of the table are not very interesting to us.

3.2 Genuine Tables

Genuine tables are those tables where the relationship between the attributes and values could be semantically established. In general, genuine tables have the following characteristics.

1) The table contains more than two cells and
2) BORDER attribute value of <table> tag can be more than one.

In this paper we only focus on genuine tables, because they include valuable knowledge and data and thus are regarded as a sort of database.

3.3 Attribute-value Pairs Extraction

Our information extraction from web tables is in the form of *attribute-value* pairs, like [2].

In contrast to most of the work on mining the web tables (except [1]), attribute and value are treated as collections. As mentioned in [3], an *attribute* consists of one or more labels, a *label* being comprised of a group of words. A *value* on the other hand, consists of one or more *quantities*.

IV. WEB TABLE ANALYSIS

In this section, based on the following assumptions we are presenting the analysis of web tables. The probability of a table containing the information row-wise (i.e. One-dimensional Table) is maximum in web tables. Because normally it is very convenient to display the table information row-wise, which ultimately increase the user's perception of data. For example, in [4], we can get the filings submitted by various companies to SEC (Security Exchange Commission). Here, the tables contained in SEC web pages are organized row-wise, where the first row is the header row. Therefore we considered the first row of the table as the heading row, so that each column will refer to the collection of a single attribute values. So the assumptions can be stated as follows:

a) The table is organized row-wise and the rows of the tables are classified into heading row and content rows. The first row of the table is always

the heading row and rest of the rows contains the data.

b) Tables that we are considering here are one-dimensional tables. Additionally, the one-dimensional tables can be further classified as row-wise, column-wise or mix-cell tables.

4.1 Heading-Row and Content-Rows Identification

The table is first analyzed to distinguish between the header row and the content rows. The header row defines to be a row that contains global information of all content rows. Primarily a heading explains the quantities in the columns. It is not that much easy to distinguish between the header rows and the content rows. In general, the header row can be identified by following rules:

1. The content of the header cells is not a quantity. In other words, from the programming view, the header cells should not contain the numbers and it should contain strings.
2. The header rows are normally the top most rows of the table. Other top most rows may contain spaces or some may contain Non-breaking Spaces (in HTML).
3. The header row is visually different from content rows. This happens when the table contents are graphically coloured or decorated by fonts and other styles.
4. The header row contains significantly fewer cells per row [5].

However the above said rules are not complete set of rules to distinguish the header row from content rows. Some more rules are to be designed for complex tables. Also, sometimes in few cases it depends on the format of tables.

Most of the time some tables are in standard format and the possible cell contents of the header row are known in advance. This is the case for the tables given on the web pages of filings on SEC website [4]. In such cases, it becomes very easy to identify the header rows, which can be identified by matching certain strings with the cell contents. The string to be matched should be the standard strings which are expected to occur in the header row. For Example, again consider the example of [4]. The DEF14A type of filings on SEC website contains a table named as “Executive Details” table. The minimum fields that are expected in this table are Name, Age, Position. In addition to these three fields they can give “Director Since” field. In such a case we can match the cell-contents of each row, with the expected strings. The point to be noted here is that, we are assuming a one-dimensional table which contents only a single header row. Thus there will be only one valid header row.

Considering the second point, the header row of a table is mostly at the top most position of the table. Some rows that may be above the header rows may contain blank spaces or Non-breaking Spaces (in HTML). Such unnecessary rows should be neglected, which can be identified by scanning the cell-contents for spaces like Non-breaking Space. Our third rule compares the visual characteristics of the header row with that of a typical row.

As stated in [5], the most reliable approach to distinguish the heading from content rows is the cell count. We can use the rule followed in [5] that if the cells count in the row being considered is equal or less than 50% of the average cell count, then it is a heading. Otherwise it is a content row.

After identifying the heading row, we will keep the record of those cell-contents in a set or a collection. We will consider this information later to identify the attribute-value pairs from remaining rows.

4.2 Row Span and Column Span

Row and Column spanning is often used in HTML in order to allow a cell to occupy more than one row or column. By default, a cell occupy (both row-wise and column-wise) only a single row or column. A positive integer is associated with the cell attributes which decides the number of rows or columns occupied by the cell. For example “rowspan=4” means the cell spans 4 rows consecutively, starting from the current column.

For identifying the attribute-value pairs in a table where for some cells the value of the rowspan or colspan attributes is greater than “1”, the simplest way is to duplicate the cell contents to each row or column it spans.

For example, again we will consider the tables given in filings on SEC [4] website. If we consider the DEF14A filings on SEC, we can get a “Summary Compensation” table. For this table, the column is always labeled with the attribute “name”. For a single name, usually there are more than one entries associated with that name attribute. This is because the value of the “rowspan” attribute of the cell containing “name”, is more than one.

4.3 Attribute-Value Relationship

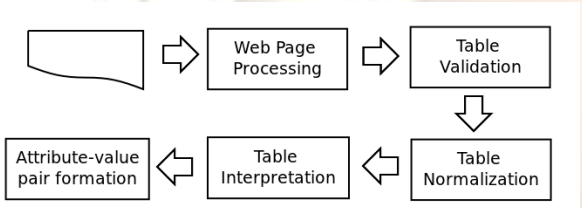
In analyzing the one-dimensional web tables, it is normally the case that the attribute and the

corresponding values are placed in a single column. Thus, we can say that the collection of values from a single column represents the complete data set for the attribute to which the column belongs. This assumption becomes very easy when we apply Normalization as stated in Section 5.

V. FLOW OF TABLE EXTRACTION

The flow of extracting the information from tables is shown in Fig. 1. It consists of five modules. The first module is Web-page processing module. This module analyses HTML text from web-page and extracts the table tags. Table Validation module validates and filter out the genuine tables using heuristic rules, neglecting the unnecessary tables. The filtered tables are sent to the Table Normalization module where the row and column spanning is normalized to a single span and duplicate the cell-contents. The Table Interpretation module distinguishes between the header-row and content-rows and identifies the valid rows and the valid cells in that row. The final tackles how to establish the relationship between the attributes and the values on the tables resulting in the formation of attribute-value pairs.

Figure 1. flow of table extraction



5.1 Web Page Processing

The source document in html format is input to the system. In this step the source document is scanned up to the end to identify the <table> tags. We can do this in two ways. The first way is to form the patterns for identifying the <table> tag and by matching the patterns we can have the tag contents. Another way is to use standard html parsing libraries like jericho [9], html parser [10]. The process of tag identification, information extraction etc becomes very easy when we use html parsers. The matter that which way is most suitable and efficient way is beyond the scope of this paper.

5.2 Table Validation

After identifying the <table> tag in step 1, we need to validate the table that is to consider only the genuine table (as defined in 3.1 and 3.2). Wang et al. [11] have proposed an algorithm for table validation. The algorithm distinguishes type of tables whether genuine table or use of page layout in HTML documents in 95% accuracy.

We considered two cases for table validation. The first case is when the expected attribute strings are known in advance. This case is applicable when the tables are given in a standard format or the table can be near to standard format. In this case the desired strings are used as keys to search for genuine table. As stated in section 4.1 we could find the genuine table by identifying the valid header row.

In second case, the format of the table is not known to us. In this case we can consider the following heuristics for table validation.

a) The table is N by M table where either M or N or both M and N are greater than two.

b) If both N and M are two or N or M is one, then the table is regarded as a list and thus no further recognition is needed.

c) If the table is N by 2 or 2 by M, where N or M is greater than 2, then the table consists of valid attributes-value pairs and thus the table is genuine one.

5.3 Table Normalization

Once a table is identified as a valid table, the process of Table Normalization is applied to the table. Table Normalization eases the Table Extraction process to extract the exact attribute-value relationship. Though we are doing this step after table validation, one can also do it before the validation.

In Table Normalization we deal with the two attributes of a cell in table viz. rowspan and colspan. As stated in Section 4.2, if the attribute value of the rowspan or colspan of a cell is greater than 1, then the cell-contents are just replicated to that many consecutive rows or columns. Table 1 show a table which has both rowspan and colspan options that combine more than two cells.

We normalize the table as shown in Table 2. After applying this normalization, we become able to treat each cell in similar manner.

Table 1. A table with rowspans and colspans

Name	Year	Salary	Bonus
Steve	2011	567,234	1,276
	2010	492,700	1,670
	2009	472,115	1,700
Michael	2011	492,115	1,875
	2010	492,115	1,875
	2009	450,000	1,738

Ronald	2011	492,115	1,875
	2010	450,000	1,738
	2009	492,115	1,875

Table 2. Table after Normalization

Name	Year	Salary	Bonus
Steve	2011	567,234	1,276
Steve	2010	492,700	1,670
Steve	2009	472,115	1,700
Michael	2011	492,115	1,875
Michael	2010	492,115	1,875
Michael	2009	450,000	1,738
Ronald	2011	492,115	1,875
Ronald	2010	450,000	1,738
Ronald	2009	492,115	1,875

5.4 Table Interpretation

The Table Interpretation is based on some heuristic rules. The attribute-value relationship can be interpreted in row-wise or column-wise manner. Here we are considering the most commonly used one-dimensional tables where data is arranged in row-wise fashion. The problem is trivial when the table tags under consideration do not contain ROWSPAN or COLSPAN. The first row consists of the attribute cells and the rest of the rows consist of the value cells. The similarity between the two rows or two cells guides us to read the table. We assume that two rows are similar if most of the corresponding cells between the two rows are similar. We can also use the Recognition algorithm proposed in [12] for checking cell similarity where each cell is represented as a vector consisting of a number of features. Also, we assumed that the header row is always at the top most positions of the table and it is the first valid row of the table.

A simple table interpretation algorithm is shown below. We assume that there are x rows and y columns. Also, let $cell_{i,j}$ denote a cell in i^{th} row and j^{th} column.

1. Considering the basic condition, if there is only one row, then we can say that the table does not contain any data. If contains the data, because only one row is present it does not contain the header-row to identify the attributes. Such table needs to be discarded, because we can't extract the attribute-value pairs from it.
2. If there are two rows then the problem becomes very easy. The first row is treated as header row and the second one as the content row. Otherwise, we start the cell similarity checking from the first row in step 2.
3. For each row i ($1 \leq i \leq x$), compute the similarity of the two rows i and i+1. If i = x, then compute the similarity of i and i-1 and then stop the process.
4. If the i^{th} row is empty, then go for the next pair of rows, i.e. i = i+1.
5. If the i^{th} and $(i+1)^{th}$ row are not similar and $i \leq (x/2)$, then the i^{th} row is treated as header row, in other words the i^{th} row contains the attribute cells. Store the labels of attributes in a list and index each label with position in row. Count the number of valid data cells in header row. After identifying the i^{th} row as header row, we will continue to find the content rows only.
6. If the i^{th} and $(i+1)^{th}$ row are similar and also both the rows are non-empty, then count the number of valid data cells in both rows. If both the counts are equal or approximately equal to the valid data cells count of header row, then both rows are treated as the content rows. Store the cells content of each row in a list indexed with their position in row.

5.5 Attribute-value pair formation

Based on the analysis stated in Section 4.3, we can conclude that in one-dimensional table the cells belonging to a single column forms a complete set of values representing a single attribute. Thus it is clear that after the normalization of the table, the index of each data cell belonging to a single attribute is same as that of the attribute label index.

Therefore, the attribute-value pairs can be formed by simply comparing index of each value in content row list with the index of attribute labels in header row list. The set of attribute-value pairs will represent a mapping from labels in header-row to their corresponding values in content-rows. For example, for the first row in Table 3 the mapping will be as follows.

{<Name: Steve>, <Year: 2011>, <Salary: 567,234 >, <Bonus: 1,276>}

VI. CONCLUSION

In this paper we propose a systematic way to extract information from HTML tables. Web Page processing, table validation, table normalization, table interpretation and attribute-value pair formation are discussed. We also propose a table interpretation algorithm which captures the attribute-value relationship among table cells. We used the cues from HTML tags and information in table cells to interpret and recognize the tables.

REFERENCES

[1] Y. Wang, and J. Hu, A Machine Learning Based Approach for Table Detection on The Web, In Proc. *11th International World Wide Web Conference, Honolulu, HI, May 2002*, pp. 242-250

[2] H. Chen, S. Tsai, and J. Tasi, Mining Tables from Large Scale HTML Texts, In Proc. *18th International Conference on Computational Linguistics*, Saabrucken, Germany, *July 2000*

[3] Yingchen Yang Wo-Shun Luk, a Framework for *Web Table Mining*, *WIDM'02, November 8, 2002*, McLean, Virginia, USA.

[4] www.sec.gov

[5] Yingchen Yang, Wo-Shun Luk, School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada, *"A Framework for Web Table Mining", WIDM'02, November 8, 2002*, McLean, Virginia, USA.

[6] V. Crescenzi, G. Mecca, and P. Merialdo. Automatic web information extraction in the roadrunner system. In Proceedings of the *International Workshop on Data Semantics in Web Information Systems (DASWIS-2001), 2001*.

[7] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards automatic data extraction from large web sites. In Proceedings of the *27th Conference on Very Large Databases (VLDB)*, Rome, Italy, 2001.

[8] C. H. Chang and S. C. Lui. IEPAD: Information Extraction based on Pattern Discovery. In *10th International World Wide Web Conference (WWW10)*, Hong Kong, 2001.

[9] http://jericho.htmlparser.net/docs/index.html

[10] http://htmlparser.sourceforge.net/

[11] Yalin WANG and Jianying HU. 2002. A machine learning based approach for table detection on the web. Proceedings of the *Eleventh International World Wide Web Conference (WWW2002)*, pages 242–250, 5.

[12] Hidetaka, Shuichi and Hiroshi, Tokyo, Recognition of HTML Table Structure, www.r.dl.itc.utokyo.ac.jp/~nakagawa/academic-res/IJCNLP04.pdf