Mining Tables on the Web for Finding Attributes of a Specified Topic

Koichi Kise Nobuhiro Ohmae
Department of Computer Science and Intelligent Systems,
Graduate School of Engineering, Osaka Prefecture University
kise@cs.osakafu-u.ac.jp

Abstract

Finding attribute-value pairs from a huge collection of HTML pages is a fundamental task for information extraction from the Web. This paper presents an unsupervised method of mining Web tables for finding attributes of a topic specified by the user. The proposed method is based on the assumption that the occurrence of text strings representing attributes is biased to the first rows and columns in tables. The χ^2-test is employed to find attribute candidates based on the assumption. Identification of attribute rows and columns using the candidates enables us to improve the accuracy of extraction. The experimental results using 2,032 pages show 81.5% precision with 67.9% recall.

1. Introduction

The explosive growth of the Web requires intelligent access to its contents for finding useful information to the user. Information extraction is one of such technologies which allows the user to access contents represented as attribute-value pairs, or in a tabular format. Although the original form of information extraction is for natural language text, some researchers have attempted to obtain information by mining entities in a more restricted format such as tables and items [1, 2, 6, 3]. We are also concerned here with the information extraction from tables on the Web.

Existing methods of information extraction from Web tables can be classified into two types: methods that requre learning with labeled data (supervised methods) [2, 3], and those that do not require such data (unsupervised methods) [1, 6]. Although the supervised methods would be more accurate for the analysis of tables about predetermined topics, they are not applicable to a new domain without intervention of the human operator. The unsupervised methods have an advantage from this viewpoint. However there exists a room for improvement of their generality, since they are based either on the predetermined types of tables [6] or the similarity measure that requires training [1].

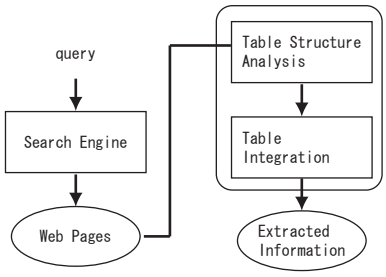

Figure 1. System architecture.

We are now developing a fully unsupervised system of information extraction from Web tables. The final goal is the query-based information extraction, i.e., to extract information *on demand* in response to the user's query. In this paper we propse a new unsupervised method of finding attributes from Web tables, which is an important part of the whole system. As compared to existing unsupervised methods, the proposed method is much simpler but yet effective as well for finding attributes from tables about various topics of interest.

2. Query-Based Information Extraction from Tables on the Web

Most of the existing methods of information extraction assume that the topic is predetermined for training of rules of extraction. Thus they are not applicable to the query-based information extraction. In order to solve this problem we focus on Web tables. Tables are the structure which represents information of objects of a topic using attribute-value pairs. Thus the task of information extraction can be rephrased to finding from Web tables all objects of the topic each of which is described as a set of attribute-value pairs

Figure 1 shows the architecture of the system which takes as input web pages obtained by a search engine and extracts information about the topic with the two steps of

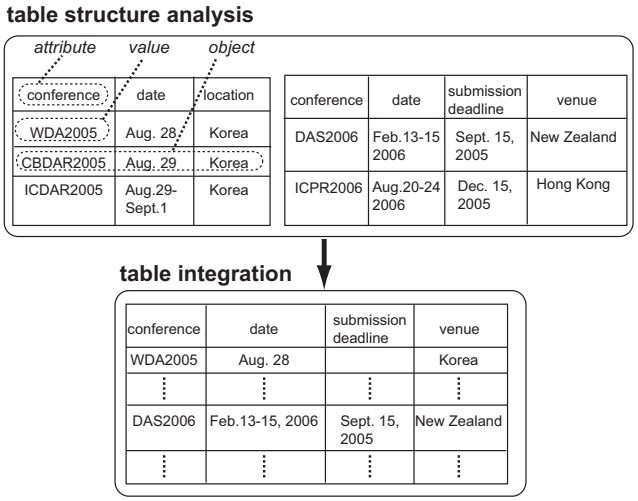

Figure 2. An example of information extraction from Web tables.

processing: table structure analysis and table integration.

Let us briefly explain each step using a simplified example shown in Fig. 2. In the following, the term *cell* is used to refer to a field of the table, and a *text string* refers to the contents of a cell if it is a text string (the contents except for text strings are meaningless to our current purpose). Each text string corresponds to either an *attribute* or a *value* of an attribute. The rows and columns which represent attributes in the table are called *attribute rows* and *columns*, respectively. The collection of cells which corresponds to a single entity is called an *object*.

The first step "table structure analysis" is to analyze tables to find attributes and values of the topic as well as to idenify objects in the table as in Fig. 2. The second step "table integration" is to merge all analyzed tables to obtain one large table that includes all the retrieved information about the topic. In order to realize this processing it is necessary to identify same attributes, values and objects which may be represented as different text strings. Thus the second step is closely related to "schema matching" that has been extensively discussed in the field of databases [4].

3. Finding Attributes

3.1. Task

In this paper we focus on a part of the first step "table structure analysis". The task here is to find attributes from the tables. The details are as follows: Given Web tables retrieved by a query, find attributes that describe the topic specified by the query without using any additional information such as "training data". In other words, the task is unsupervised extraction of attributes. For the example in Fig. 2, the query would be "conference", and the attributes that should be extracted are "conference", "date", "venue" and "submission deadline". The task is not trivial due to the following reasons:

- Retrieved tables are not necessarily *genuin* tables, but often include a lot of tables just for layout [5].
- Not all first rows and first columns represent attributes. The position of rows and columns which represent attributes vary depending on tables.

3.2. Extraction of Attribute Candidates

Our idea for accomplishing this task is based on the assumption that the location of text strings representing attributes is *biased* to first rows and columns. In other words, a text string seems to represent an attribute if it exists more frequently in either first rows or columns as compared to other fields.

The following processing is applied to each text string in the tables. In order to evaluate the bias of the location, we employ a statistical test called χ^2-test of goodness of fit. The null hypothesis for the test is that the location is unbiased. Let f_1 and f_2 be the observed frequencies that the text string is in the first rows and columns, and in cells except for the first rows and columns, respectively. Note that the frequency is counted in all available tables. Let np_1 and np_2 be the expected frequencies for the first rows and columns, and for the remaining cells, rescpectively. The values are calculated as follows. Suppose we have a 3×5 table. If the occurrence is unbiased, the text string exists in each cell with the equal probability $1/15$. The number of cells for the first rows and columns is 7 and thus the expected frequency for the first rows and columns is $7/15$. By summing up the values for all tables, we can obtain np_1. The value of np_2 is similarly calculated. Based on the above preparation, the value of χ^2 with Yates correction is calculated by:

$$\chi^2 = \frac{(|f_1 - np_1| - 0.5)^2}{np_1} + \frac{(|f_2 - np_2| - 0.5)^2}{np_2} \,. \quad (1)$$

Let $\chi^2_\alpha(1)$ be the value in the χ^2 table for the degree of freedom of 1 and the significance level α. If $\chi^2 > \chi^2_\alpha(1)$, the null hypothesis can be rejected. In other words, the text string can be considered to represent an attribute. The text strings regarded as attributes at this step are called attribute candidates in the following.

3.3. Identification of Attribute Rows and Columns

In general it is hard to extract all possible attributes by only the previous step of χ^2-test, since some of the at-

Table 1. Queries.

id	query	language
1	gakkai(conference), kaisai(hold)	Japanese
2	yakyu(baseball), senshu(player)	Japanese
3	international, conference, date, location	English
4	baseball, player, roster	English

Table 2. Statistics of the data.

id	# pages	# tables	# col.	# rows
1	672	11,645	22,587	47,736
2	753	11,581	25,901	34,174
3	251	2,354	4,580	15,575
4	356	3,498	8,560	15,403

tributes occur too less frequently to reject the null hypothesis. This step of processing for identifying attribute rows and columns is applied to find such attributes. The processing is quite simple. All attribute candidates are located in each table. If a row (column) includes a large enough number of candidates, it can be regarded as an attribute row (column). Note that in this processing, rows and columns are not necessarily the first rows and columns. In addition, multiple rows or columns can be selected as attributes. Once attribute rows and columns are identified, all the text strings on the attribute rows and columns are identified as attributes even if they are not the attribute candidates.

4. Experimental Results

4.1. Data

The proposed method was tested using several queries both in Japanese and English as listed in Table 1. With the help of Google API, top 1,000 pages were downloaded from the Web. We obtained pages for experiments by deleting PDF pages from the downloaded pages. Table 2 shows the statistics of the pages for experiments.

4.2. Results of the Extraction of Attribute Candidates

We first evaluated the effectiveness of the extraction of attribute candidates. The significance level was set to 0.1%. Results were evaluated using recall R, precision P and their harmonic mean called F-measure F, all of which were measured in the number of attributes.

The results are shown in Table 3. Because the significance level α is low, the selection was conservative: higher precision was achieved with lower recall. This is important to the proposed method since the attribute candidates selected at this step serve as clues to find remaining attributes at the next step.

Table 3. Results of the extraction of attribute candidates.

id	Recall R	Precision P	F
1	37.2% (55/148)	65.5% (55/84)	47.5%
2	47.1% (49/104)	58.3% (49/84)	52.1%
3	37.3% (66/177)	89.2% (66/74)	52.6%
4	20.3% (31/153)	70.5% (31/44)	31.5%
ave.	35.5%	70.9%	47.3%

Erroneously extracted candidates can be classified into two types: (Type 1) Candidates were attributes but for different topics (64% of the total errors); (Type 2) Candidates were not attributes but attribute values (36%). Type 1 errors included “days of the week” on calendars in blogs, which were observed for the query id 2. Type 2 errors were caused by attribute values that were frequently on the first columns or rows due to their commonality in many objects. Errors as not selecting correct attributes were mainly due to their small frequency (typically only once) in all tables.

4.3. Results of the Identification of Attribute Rows and Columns

The next step is the identification of attribute rows and columns. First, tables that include attribute candidates were selected. Note that this processing naturally filtered out tables only for the layout purpose. No layout tables were erroneously selected at this step. Next, a simple rule for the identification was applied to the remaining tables. The rule was that attribute rows and columns should include more than or equal to two attribute candidates.

Figure 3 illustrates the results. In this figure, the left circles indicate the numbers of attribute candidates extracted at the first step. The right circles represent the numbers of attributes identified at this second step. The upper parts of the circles are for correct attributes, while the lower parts are for incorrect ones. As shown in this figure, the step of identification worked quite well: a lot of incorrect attributes were eliminated and a large number of correct attributes were added while the number of newly added incorrect attributes was kept small.

Erroneously extracted attributes can be classified into two types: (Type 1) Errors caused by incorrect attribute candidates (57% of the total), (Type 2) Errors caused by this step of identification (43%). Type 2 errors were due to irrelevant tables that included correct attributes in their rows or columns.

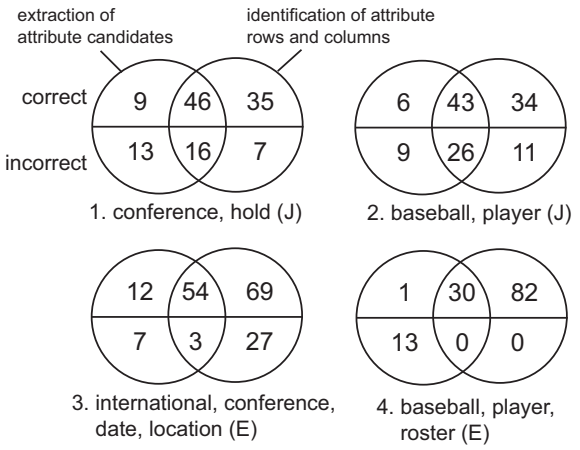

Figure 3. The numbers of correct and incorrect attributes at each step of processing.

Table 4. Results after the identification of rows and columns.

id	Recall R	Precision P	F
1	54.7% (81/148)	77.9% (81/104)	64.3%
2	74.0% (77/104)	67.5% (77/114)	70.6%
3	69.5% (123/177)	80.4% (123/153)	76.1%
4	73.2% (122/153)	100% (112/112)	84.5%
ave.	67.9%	81.5%	74.1%

Table 4 shows the results in the same format as in **4.2**. As shown in the table, not only the improvement of recall but also that of precision were achieved by this step of identification. Recall was improved by finding attributes that were not selected as candidates but were in the same rows or columns with the correct candidates. Precision was raised by eliminating incorrect candidates that were not in the same rows or columns with other candidates.

Table 5 lists recall and precision measured in the number of not attributes but tables. For the query id 3, for example, 334 tables were with the identified rows or columns, and 99.1% of which were tables relevant to the query. In total 397 correct tables were included in the dataset and 83.4% of which were identified. For the query id 2, a lot of errors were simply caused by the calendars in blog pages, which can be easily eliminated by preprocessing. The brackets indicate the numbers without such blog pages.

In order to improve further the processing accuracy, it is necessary to improve the effectiveness of identification of attribute rows and columns. One possibility is to take into account certainty of not only being attributes but also being attribute values in the identification step.

Table 5. Results measured in the number of tables.

id	Recall R	Precision P	F
1	79.0% (147/186)	59.0% (147/249)	67.6%
2	75.0% (123/164)	25.4% (123/485)	37.9%
		[84.2% (123/146)]	[79.3%]
3	83.4% (331/397)	99.1% (331/334)	90.6%
4	64.7% (55/85)	100% (55/55)	78.6%
ave.	75.5%	70.9%	68.7%
		[85.6%]	[79.0%]

5. Conclusion

We have proposed a method of extracting attributes that describe a specified topic by analyzing Web tables. The characteristic points of the method are as follows. (1) It employs χ^2-test for extracting attribute candidates based on the assumption that the occurrence of attributes is biased to the first rows and columns in tables. (2) Additional attributes are extracted by identifying attribute rows and columns based on the attribute candidates. From the experimental results on 29,078 tables, 67.9% recall and 81.5% precision were obtained for various attributes.

Future work includes the development of the system of query-based information extraction which employs the proposed method as a module.

References

[1] H.-H. Chen, S.-C. Tsai, and J.-H. Tsai. Mining tables from large scale html texts. In *Proc. 18th Int'l Conf. on Computational Linguistics*, pages 166–172, 2000.

[2] W. W. Cohen, M. Hurst, and L. S. Jensen. A flexible learning system for wrapping tables and lists in html documents. In *Proc. WWW2002*, 2002.

[3] K. Itai, A. Takasu, and J. Adachi. Information extraction from html pages and its integration. In *Proc. 2003 Symposium on Applications and the Internet Workshops*, pages 276–281, 2003.

[4] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[5] Y. Wang and J. Hu. Automatic table detection in html documents. In A. Antonacopoulos and J. Hu, editors, *Web Document Analysis*, pages 135–154. World Scientific, 2003.

[6] M. Yoshida, K. Torisawa, and J. Tsujii. Extracting attributes and their values from web pages. In A. Antonacopoulos and J. Hu, editors, *Web Document Analysis*, pages 179–200. World Scientific, 2003.