A PROTOTYPE OF SEARCH ENGINE FOR TABLES ON THE WEB

Noguchi, Masato
Kyushu University Graduate Scool of ISEE

Hirokawa, Sachio
Computing and Communication Center, Kyushu Unviersity

出版情報：Proceedings of International Symposium on Information Science and Electrical Engineering. 2003, pp.561-564, 2003-11
バージョン：
権利関係：

A PROTOTYPE OF SEARCH ENGINE FOR TABLES ON THE WEB

Masato Noguchi

Kyushu Univ.
Graduate School of ISEE
Hakozaki 6-10-1, Fukuoka 812-8581, JAPAN

Sachio Hirokawa

Kyushu Univ.
Computing and Communication Center
Hakozaki 6-10-1, Fukuoka 812-8581, JAPAN

ABSTRACT

There are huge amount HTML pages on the Web. Many of them contains lists and tables. It is often the case that a line represents an instance of a record and that each column represents an attribute of the record. Given query words, similar words can be obtained from a column if the query words are contained in the column. Given a record, as a pair of words, similar pairs can be obtained from other lines of the table. Therefore, the information of tables enables search of similar words and similar records. The granularity of these search is more fine compared to the conventional search engines whose result is a list of HTML pages.

We are developing a search engine for tables on the Web. It detects lists and tables by analyzing repeated structure in a HTML page. No examples are necessary in advance. No interaction is required in processing. No special knowledge or no natural language processing are necessary to detect the similarity of the data.

This paper describes the structure of the system, the indexing mechanism of tables and typical applications of fine search using table index.

1. INTRODUCTION

This paper reports basic ideas and the key features of a search engine for tables on the Web[1, 2]. There are huge amount HTML pages on the Web. Many of them contains lists and/or tables. It is often the case that items of the same kind appear in the same column or in the same row in a table. Therefore, by using information on a table, it becomes possible to collect data of the same kind efficiently. Moreover, the granularity of the collected data is "fine" compared to the conventional search engines whose result is a list of HTML pages.

Search engines are indispensable tools for utilizing WWW. Imagine that you are looking for information on personal computers and comparing the price of them each other. You are not looking for the Web pages which explains the detail informations of specific PCs of a company. You are looking for good lists where you can glance them and compare them. Conventional search engines returns pages of computer makers and computer shops. But the granularity of the pages vary from page to page. Even if you would find Web pages where some kind of listings are provided, you need to read the pages and need to create your own list out of the listings.

Quality and quantity have been two main measure for evaluating and comparing Web search engines. But too much search results are beyond the comprehension of users. To cope with large number of search results, Google[1], e.g. strengthened the ranking of the results, Yahoo![2] provided directory of the Web pages, Vivisimo adapted its clustering technology and Kartoo[3] introduced a unique visual interface. Another direction to guarantee the quality is to focus on specific area. Siteseer[4] is a search engine for scientific literature and FlipDog[5] is a search engine for jobs, both of which apply innovative technologies to collect and extract specific data on the Web. These two companies are being differentiated by speciality nature. We consider that the key feature of these new trends is not in the speciality but in the granularity of search results. Users are not looking for Web pages, but are looking for more fine contents of pages. We propose the granularity as a new measure for search engines. We describe our technology as a solution for fine search engine of next generation.

We are developing a prototype of SABLE - a search engine for tables on the Web. The SABLE system discovers and collects HTML pages which contain tables. There have been many research for discovering tables on the Web Page. Some of them consider only tables described by TABLE tags of HTML. [3] uses item features, such as numerical values, the names of a place, and a name of a person. [4] requires a user to prepare a template for target pages. In [5], many examples are necessary for learning to detect genuine tables.

SABLE detects lists and tables by analyzing repeated

[1] http://www.google.com
[2] http://www.yahoo.com
[3] http://www.kartoo.com
[4] http://citeseer.nj.nec.com
[5] http://www.flipdog.com

structure in a HTML page. Therefore no examples are necessary in advance. No interaction is required in processing. The SABLE system recognizes the table structure and extracts the contents in the same row or in the same column as data of the same kind. These data are more fine compared to original HTML pages. The items on the same column, or in the same row, can be used as keywords for searching for next tables of similar contents. This process accumulates many tables of the same contents if it is applied repeatedly. The strength of our method is in simplicity. No special knowledge or no natural language processing are necessary to detect the similarity of the data.

This paper describes the structure of the system, the indexing mechanism of tables and typical applications of fine search.

2. TABLE EXTRACTION AND FOCUSED SEARCH

A search engine requires a huge database of HTML pages. Construction of such a database for general search purpose is not in the scope of the present paper. The search engine we are constructing is not a general search engine, but a search engine specialized for tables on the Web. We do not want collect any web pages but pages with tables or lists of specific fields. There are two problems to achieve our goal. The first problem is to discover tables on the web. The second problem is to control explosion of crawling so that collected pages are focused on the subjects.

Table detection and extraction

We apply a method [1] to detect and extract table structure from an HTML file. The method discovers repeated patterns in the tree structure of an HTML file. An instance of the same repeated pattern are regarded as a line of a table. Thus, we construct tables from an HTML file.

Focused Search

The table detection method does not depend on how the pages are collected. But collecting whole data on the Web is not our purpose. A robot or a crawler generally tend to collect web pages regardless of the contents. One of the difficulties is how to get rid of explosion of crawling. We should focus on tables and should collect tables with similar contents. So, we want get rid of collecting pages with no tables. Even if a page contain a table, we prefer collecting large cluster of tables of similar contents to collecting very few tables of rare contents.

We propose a simple solution for focus search using the index information of tables. Imagine that you are collecting a list of hot springs. You would search a general search engine with a keyword “hot spring”, and you would acquire several names of specific hot springs. If two of them appear on the same column in a table, the chances are that the data on the same column are names of hot springs as well. We can expand a list of hot springs from two or three examples using table information. Although this idea may look too simple and naive, but we confirmed that similar words can be collected in large quantities without explosion of crawling. In section 4.2, we describe the method in detail.

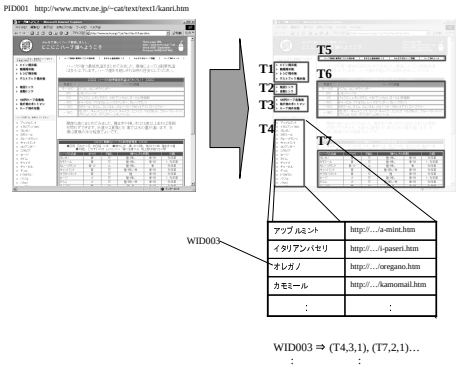

Fig. 1. Extracting process and unique ID

3. INDEXES OF TABLE INFORMATION

The collected pages are analyzed and the results are stored in a database. In the conventional search engines, units of database are pages and keywords. It keeps which page contains which keywords (index) and which keyword occurs which pages (inverted index). On the other hand, tables are core information unit in our system. Fig.1 shows the progress of extraction of a table, where each object – a Web page, a table, and a word – is discriminated and dealt with by assigning ID. Hence, a table in a page is designated by (P, T) with the index P of the page and with the index T of the table in the page. A column is indexed by (P, T, c) and a position of a table is indexed by (P, T, c, r) where c and r are indexes of the column and row.

3.1. Page ID and Page Information

There are various kinds of information that we refer as page information on a web page. The page information of a page may contain the page ID, URL, title and the list of table id’s included in the page. To search pages, the list of keywords that appear in the page is important information of the page. We prepared two indexes to obtain a page information. One is the “page information index” for searching page information by Page ID, and another is the “inverted page information index” for searching Page ID by URL. Accordingly, if Pages ID or URL are known, all page information of that page can be known.

3.2. Table ID and Table Information

An acquired Web page is analyzed as a tree structure. Then tables included in the page are extracted and unique ID is assigned for each table. At the same time, the strings in the pages are assigned unique IDs as well. Note that the same string have the same ID even if it appears in different positions of different tables in different pages.

A table has its own information as a page has its own information. A table information consists of the table ID, the table size, the position on the page and the list of words in the table. In practice, the list of words is not a simple list of words, but a list of (word, column, row)'s which means that the word appears at the position (column, row) of the table.

Moreover, the table information as well as page information has the forward index, and the inverted index. The table index is for searching table information by table ID, and the inverted index is for searching list of positions where the word appears.

3.3. Word ID and Word Information

Word information associates a string with the word ID. There is "word index" for searching string by word ID, and "inverted word index" for searching word ID by string. We implemented two type of string matching – the exact matching and the partial matching. The exact matching is realized with hashing, the partial matching is realized by bigram.

4. SEARCH WITH FINE GRANULARITY

The indexes of our system have several granularities – page, table, column, row, position and word. There are various search can be performed from various viewpoints. The granularity of index makes a big difference of our search system to the conventional search engine which returns a list of URLs for query words. We can classify searches according to the unit of information. We demonstrate three kinds of searches which utilizes indexes of fine grain. The first example is the simple table search, where (P, T, c, r) is the unit and tables are obtained for given a query word. The second example is the similar word search, where (P, T, c) is the unit and similar words are obtained for given words. The third example is the record search, where (P, T, c_1, c_2) is the unit and return a list of records or the specified record. In all examples, a received query is transformed to the word ID at the beginning. Then positions of the word is searched in the inverted index.

4.1. Simple Table Search

The first example is to search tables which contains the query.

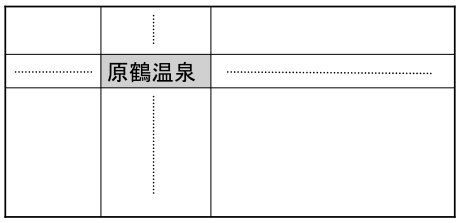

Fig. 2. Simple Table Search

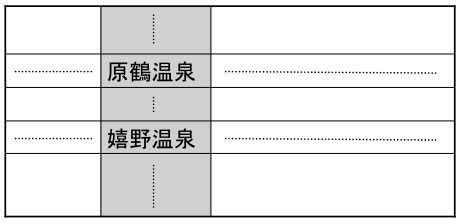

Fig. 3. Similar Word Search

> Find tables which contains the word "原鶴温泉 (Harazuru spa)".

A table obtained with this query is as it is shown in Fig.2.

No one can tell correctly the intention of the user who demanded this query. But chances are that he is searching for information related to Harazuru spa ,for example, such as a location, a contact or a spa list of common feature with Harazuru spa. When the query of this form is received, the system transform the query into the word ID, and looks up the inverted index and returns a list of tables which contains the query word.

4.2. Similar Word Search

The intention of similar word search is more clear than that of simple table search. A user is assumed to be looking for similar words in this search. We consider that the contents of the same column of a table represents the same feature. If query words are found in the same column, the other contents in the column may be similar words.

> Find similar words as "原鶴温泉 (Harazuru spa)" and "嬉野温泉 (Ureshino spa)".

Fig.3 shows a target table, and a target column. By using query of this form, a user can collect the list of two or more words similar to specified words. This list is useful than a simple list, because the other column provides additional information to each word.

In the actual implementation, a system compares the po-

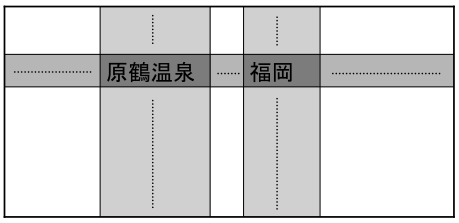

Fig. 4. Record Search

sition of the query words in each target table. Since position information is formed as table-row-column, this process is performed quite easily.

4.3. Record Search

A search of record is implemented as follows. A basic idea is to consider a line of a table as an instance of record. In Fig. 4, "原鶴温泉 (Harazuru spa)" is a name of hot spring and "福岡 (Fukuoka)" is the name of prefecture where Harazuru spa locates. Thus the pair of two column represents a record of spa with the name and the location. In the similar word search, we assume that query words represent the similar attribute. In the record search, we assume that query words represent different attribute of a record.

By this processing, the following questions can be answered.

> If "原鶴温泉 (Harazuru spa)" reminds me of "福岡 (Fukuoka)" , what are other pairs of words which have such a relation?

There are no big difference between the similar word search and the record search in the processing. But there is a big difference in the result returned. In the similar word search, a list of words, which is considered as similar to the query, is returned as a result. On the other hand, the record search returns a list of pairs of words. The pairs are assumed to have the similar relation as query words have.

5. CONCLUSION AND FUTURE WORK

We presented a prototype of search engine for tables on the Web. We reported the indexing mechanism of tables and typical applications of fine search using the table indexes. A table in a Web page is detected and extracted as a repeated pattern, where no knowledge and natural language processing is required. The index of tables realizes more fine search compared with conventional search engines.

There are three directions of future work. The first is to increase the number of Web pages which the database covers. The second is to apply searching for various subjects. The third is to introduce the ranking between items.

6. REFERENCES

[1] M. Noguchi and S. Hirokawa, "Collecting similar words using softpath," Tech. Rep., JSAI SIG-KBS-A203, pp.15-20, 2002.

[2] M. Noguchi and S. Hirokawa, "Knowledge aquisition of similar words from web," Symposium of Informatics, pp.21-24, 2003.

[3] S.Sato and M.Sato, "Toward automatic generation of web directories," Proc. of International Symposium on Digital Libraries 1999 (ISDL'99),pp127-134, 1999.

[4] M. Umehara, K. Iwanuma, and H. Nagai, "A case-based semi-automatic transformation from html documents to xml ones –using the similarity between html documents constituting a series–," *JSAI*, vol. 16, no. 5, pp. 408–416, 2001.

[5] Yalin Wang and Jianying Hu, "A machine learning based approach for table detection on the web," The Eleventh International World Wide Web Conference, 2002.