Semi-Automatic Construction of Metadata from A Series of Web Documents

Hirokawa, Sachio
Computing and Communications Center, Kyushu University

Itoh, Eisuke
Computing and Communications Center, Kyushu University

Miyahara, Tetsuhiro
Faculty of Information Sciences, Hiroshima City University

出版情報：Lecture Notes in Computer Science. 2903, pp.942-953, 2003-12. Springer Berlin / Heidelberg
バージョン：
権利関係：The original publication is available at www.springerlink.com

Semi-Automatic Construction of Metadata from a Series of Web Documents

Sachio Hirokawa[1], Eisuke Itoh[1], and Tetsuhiro Miyahara[2]

[1] Computing and Communications Center, Kyushu University
Hakozaki 6-10-1, Higashi-ku, Fukuoka, 812-8581, Japan
{hirokawa,itou}@cc.kyushu-u.ac.jp
[2] Faculty of Information Sciences, Hiroshima City University
Otsuka-Higashi 3-4-1, Asaminami-ku, Hiroshima, 731-3194, Japan
miyahara@its.hiroshima-cu.ac.jp

Abstract. Metadata plays an important role in discovering, collecting, extracting and aggregating Web data. This paper proposes a method of constructing metadata for a specific topic. The method uses Web pages that are located in a site and are linked from a listing page. Web pages of recipes, real estates, used cars, hotels and syllabi are typical examples of such pages. We call them a series of Web documents. A series of Web pages have the same appearance when a user views them with a browser, because it is often the case that they are written with the same tag pattern. The method uses the tag-pattern as the common structure of the Web pages.

Individual contents of the pages appear as plain texts embedded between two consecutive tags. If we remove the tags, it becomes a sequence of plain texts. The plain texts in the same relative position can be interpreted as attribute values if we presume that the pages represent records of the same kind.

Most of these plain texts in the same position vary page to page. But, it may happen that the same texts show up at the same relative position in almost all pages. These constant texts can be considered as attribute names. "Location", "Rating" and "Travel from Airport" are examples of such constant texts for pages of hotel information. If the frequency of a text is higher than a threshold, we accept it as a component of metadata.

If we mark a constant text with "N" and a variable text with "V", the sequence of plain texts forms a series of N's and V's. A page in a series contain two kinds of NV sequence pattern. The first pattern is $(NV)^n$, which we call vertical, where an attribute value follows the attribute name immediately. The second pattern is N^nV^n, which we call horizontal, where names occur in the first row and the same number of values follow in the next row. Thus we can understand the meaning of values and can construct records from a series of Web pages.

Keywords: Knowledge acquisition, Knowledge engineering, Knowledge discovery and Data Mining, Machine learning, Ontology.

T.D. Gedeon and L.C.C. Fung (Eds.): AI 2003, LNAI 2903, pp. 942–953, 2003.

1 Introduction

Due to the rapid spread of the Web, huge amounts of information in various formats are available on the Web. In order to extract useful information from huge Web pages, there are many research tasks such as extraction of knowledge from semistructured data or HTML files [2, 4, 9], topical crawling which automatically collects Web pages of user's topics [1, 3], and integration of semantically homogeneous information which is heterogeneous in representation.

We are constructing an information integration system utilizing Web pages on specific topics. In order to realize such an information integration system, we need to realize the following subtasks: collection of Web pages on a specific topic, classification of the collected Web pages, information extraction from Web pages, and construction of a database of extracted information. However, Web pages on a specific topic are available in multiple Web sites of different formats. To resolve differences between multiple databases, the database schema is necessary for each source and metadata is necessary as an integration target.

There have been many efforts to construct metadata for this purpose. But it is not a trivial task. In this paper we propose a new method for semi-automatic construction of metadata from a *series of Web pages* [14]. A series of Web pages are pages that are located in a site and are linked from a listing page in the site. Web pages of recipes, real estates, used cars, hotels and syllabi are typical examples of such pages. These series of pages can be searched with a phrase "list of" and a keyword of the topic.

In most cases, a series of Web pages are in the same site and are linked from a page of contents in the site. We call the source page of the links as a *page of type A*. The Web pages that are linked from the page of type A are called *pages of type B*

A key problem in Web Mining is the separation of the structure and the contents from an HTML file. A series of Web pages have the same appearance when a user views them with a browser, because it is often the case that they are written with the same tag-pattern. The proposed method uses the tag-pattern as the structure of the Web pages. Individual contents of the pages appear as plain texts embedded between two consecutive tags. If we remove the tags, it becomes a sequence of plain texts. The plain texts in the same relative position can be interpreted as attribute values if we presume that the pages represent records of the same kind. Most of these plain texts in the same position vary

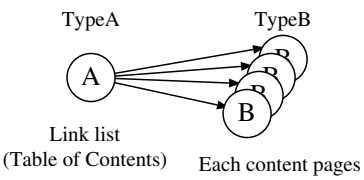


Fig. 1. Link Structure of a Series of Web Pages

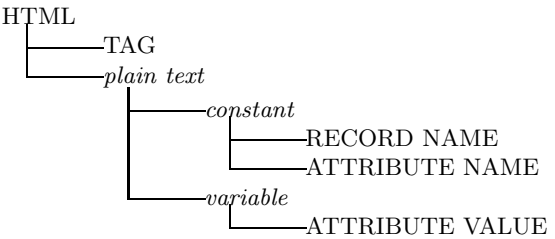


Fig. 2. Structure and Contents of HTML Files

page to page. But, it may happen that the same texts show up at the same relative position in almost all pages. These constant texts can be considered as attribute names. "Location", "Rating" and "Travel from Airport" are examples of such constant texts for pages of hotel information. If the frequency of a text is higher than a threshold, we accept it as a component of metadata. Our solution is to use (TAG, RECORD NAME, ATTRIBUTE NAME) as the structure, and ATTRIBUTE VALUE as contents (see Fig. 2).

If we mark a constant text with "N" and a variable text with "V", the sequence of plain texts forms a series of N's and V's. A page in a series contains two kinds of NV sequence pattern. The first pattern is $(NV)^n$, which we call vertical, where an attribute value follows the attribute name immediately. The second pattern is N^nV^n, which we call horizontal, where names occur in the first row and the same number of values follow in the next row. Thus we can understand the meaning of values and can construct records from a series of Web pages.

Making content-related metadata plays an important role in information integration on Web pages. As the extensive studies on the Semantic Web, RDF and RDF schema show, extraction of meaningful metadata describing the contents of Web pages is the key to realize an information integration system. However, almost all Web pages in the current WWW are HTML files and the acquisition of appropriate metadata is still a major problem to realize such a system. Hence we propose a new method for semi-automatic construction of metadata from a series of Web pages as a realistic method for implementing information integration on the Web pages.

In order to extract a template specific to a series of Web pages, our method uses a measure of structural similarity among Web pages. Measuring the structural similarity among semistructured data has been an active research topic [4, 5, 6, 8, 10] and it is fundamental to many applications such as integrating Web data sources. The authors have proposed a method for extracting a common tree structured pattern from semistructured data [11]. This extraction method can be applied to extracting a template specific to a series of Web pages.

Umehara et al. [14] targeted a series of Web pages. But their aim is not in generating metadata, but in transforming a series of HTML files into a series of corresponding XML files. Their method requires a user to prepare transformation

examples. Stuckenschmidt et al. [17] proposed a knowledge-based approach for metadata validation and generation but their approach is within a framework of ontology. Handschuh et al. [16] proposed a general framework for creation of semantic metadata in the Semantic Web but the framework does not deal with raw data of Web pages. Arakas et al. [2] proposed an extraction method of metadata from Web pages in a Web site but do not consider information integration using metadata.

These works focus on some of the subtasks such as extracting a template or making metadata. But our aim is to construct a total system of information integration utilizing Web pages on specific topics. We believe that our method will be improved by using the methods of these related works.

This paper is organized as follows. In Section 2, we explain the idea with the Web documents of hotel information. In section 3, we propose a method for extracting records from a series of Web pages by transforming the pages into tag sequences. Then the metadata is constructed as a list of strings that appear in the same position of all the records. In section 4 and 5, we give a method for combining the names and the values of attributes. In Section 6 we conclude this paper.

2 Metadata for "Hotel"

In this section, we explain the idea of constructing metadata for "hotels". An online hotel reservation site "Bookingsavings" [1] has lists of hotels for many destinations. For example, a list of hotels in Perth is displayed in Fig. 3.

The list has 20 links to the pages of detailed information of the hotels, where "Rating", "Rates", "Room facilities", "Hotel facilities", "Location", "Travel from Perth Airport", "Travel from railway Station", "Children/extra bed" and "Places of interest nearby" are displayed in the same pattern as we see in Fig. 4.

The page for "Novotel Vines Resort" contains "Wineries", "Historic Sites" and "Local Attractions" but does not contain other fields. Nevertheless, 19 pages are written with the same pattern. We describe the pattern as the following tag-sequence, where "*" represents the positions of texts which vary hotel to hotel.

```
html head title * meta meta meta meta link /head body div table tr td
img /td /tr tr td div * * * /td /tr /table table tr td br table tr td
img /td /tr /table div * div * br a img /a br br span * * img img img
/span br span * * * br br div * br div * br div * br div br li b * *
br li b * * br li b * * br li b * * br li b * * br li b * * br li b *
* br br /td /tr tr td a * * * * * * * /td /tr /table table tr td img
/td /tr /table a * /div /body /html
```

The pattern contains 36 such fields or texts. These 36 texts form a record of each hotel information. The fields are classified as common parts and individual

[1] http://www.bookingsavings.com/asia_pacific/australia/perth/index.shtml

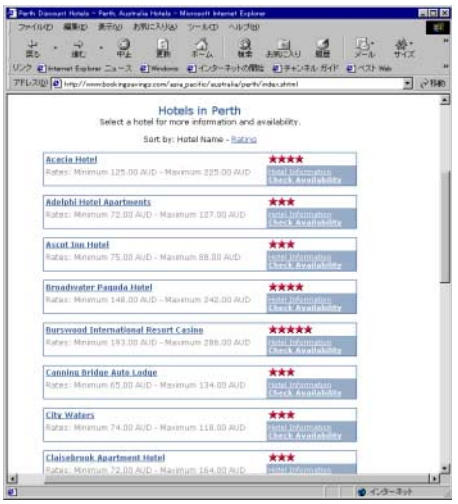


Fig. 3. A List of Hotels in Perth

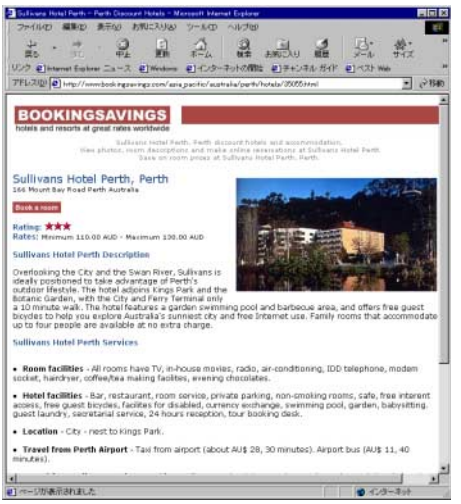


Fig. 4. A Page of Detailed Information of a Hotel

parts according to the frequency. The common parts appear as the names of fields of the record. A field with frequency 19 represents a name of an attribute. On the other hand, the texts in a field with frequency 1 vary according to each HTML page. There may be some exception, as in the 25th field of "Ascot Inn Hotel", where "Extra bed" is used instead of "Childbed/extra bed" and as in the 23rd field of "City Waters Hotel", where "Travel from Perth" is used instead of "Travel from Perth Airport".

3 Metadata Construction Algorithm

In this section, we describe an algorithm for constructing metadata given a keyword. Fig. 5 shows the algorithm.

In the first step, we send the keyword to a search engine for collecting pages related to a topic. Here we send the keyword augmented with the phrase "list of". Thus we obtain Web pages which are supposed to have many links to related pages. These are the pages of "type A" as we explained in the previous section. A page of type A contains many links to the desired pages, i.e., pages of "type B". But it may contain other kind of links, e.g., links to the top pages and links to famous pages. It is often the case that the pages of "type B" are located in the same directory in the same site. Such directory is calculated with the base URL. For example, the HTML files of the hotels in Section 2 are in `http://www.bookingsavings.com/asia_pacific/australia/perth/hotels/`. The pages which are linked from the page of type A and are in the directory are the pages of type B.

Table 1. Contents and Frequency

field	frequency	contents
1	1	Sullivans Hotel Perth - Perth Discount Hotels
2	1	Sullivans Hotel Perth, Perth discount hotels ...
3	1	View photos, ... at Sullivans Hotel Perth
4	1	Save on room prices at Sullivans Hotel Perth, Perth.
5	1	Sullivans Hotel Perth, Perth
6	1	166 Mount Bay Road Perth Australia
7	19	Rating:
8	19	
9	19	Rates:
10	19	
11	1	Minimum 110.00 AUD - Maximum 130.00 AUD
12	1	Sullivans Hotel Perth Description
13	1	Overlooking the City and the Swan River, ...
14	1	Sullivans Hotel Perth Services
15	16	Room facilities
16	1	- All rooms have TV, in-house movies, ...
17	19	Hotel facilities
18	1	- Bar, restaurant, room service, ..
19	19	Location
20	1	- City - nest to Kings Park.
21	7	Travel from Perth Airport
22	1	- Taxi from airport (about AU$ 28, 30 minutes)....
23	8	Travel from railway station
24	1	- Taxi from railway station (about AU$ 5, 5 minutes)....
25	6	Children/extra bed
26	1	- Maximum 2 children are allowed to stay ...
27	19	Places of interest nearby
28	1	Kings Park, Swan River.
29	19	Worldwide Hotels
30	19	–
31	19	Asia Pacific Hotels
32	19	–
33	19	Australia Hotels
34	19	–
35	19	Perth Hotels
36	19	Copyright, terms and conditions.

The second step is to obtain the common tag-pattern of pages of type B. It is obtained as the tag-sequence for a Web page which belongs to the maximal cluster with respect to the tag-sequence mapping. To use the tag-sequence is introduced in [12]. The tag-sequence mapping is a function from an HTML file to the tag-sequence of the file. It eliminates attributes of tags and deletes textual contents which appear outside of HTML tags. The *maximal cluster* is defined as follows. Let f be a mapping from a set X to a set Y. A *cluster* in X with respect to f is an inverse image of $y \in Y$ with respect to f, i.e., $f^{-1}(y) = \{x \in X \mid f(x) = y\}$. A cluster is *maximal* if the number of elements in the cluster is maximal.

The 3rd step is to extract the contents from pages of type B using the tag-sequence. We do not need pattern matching at this stage. Because, we already obtain the corresponding contents when we calculate the tag sequence of the HTML file. At this stage, the i-th page $a[i]$ of type B is represented as a list $a[i, 1], a[i, 2], ..., a[i, n]$ of strings. Since the pages of type B are supposed to have

the same tag sequence, the length n of this strings is the same to all pages of type B. Thus a page of type B is transformed into a record with n-fields.

The final step is to classify the fields and distinguish the names of the attributes and the values of the attributes. If all j-th fields have the same string, we consider the field represents the name of the attribute whose contents follow in the sequel. Conversely, the attribute values vary one by one. So, we distinguish the names and the values of attribute by the frequency of the strings among the same fields.

4 Alignment of Names and Values

We can consider an HTML page to be a merged sequense $(T_1, P_1, T_2, P_2, \cdots, T_n, P_n, T_{n+1})$, where T_i is a tag and P_j shows a text. A series of Web pages $H_1, H_2, \cdots, H_m$ have the same tag-sequence, so that the sequence of plain texts $(P_1^i, P_2^i, \cdots, P_n{}^i)$ can be extracted uniformly from each page $H_i (i = 1, \cdots, m)$.

The plain text P_j^i in the j-th position may be an attribute name or an attribute value. In the previous section, we showed an algorithm to distinguish names and values according to the frequency of the text. In this section and next section, we explain how to name the values or how to bind names and values.

Consider a series of pages of "used cars" as an example. Characteristic keywords of these pages are "maker", "model", "year", "mileage", "color" and so on. These keywords are the attribute names and are displayed with attribute values aligned vertically or horizontally when we see the pages with a browser (Fig. 6). In both cases of alignment, a name and the corresponding value are displayed close to each other. Such display improves the visual effect and helps user's comprehension.

Imagine that we have a series of pages as in Fig. 7 and that we know that F_1 is a name. Where does a corresponding value appear? If F_3 is another instance of name, the value for F_1 should be F_2. If F_2 is a name, then the value for F_1 should be F_3. Thus, if names are aligned vertically the corresponding value appears horizontally next to the name. If names are aligned horizontally, the corresponding value appears vertically next to the name.

5 Binding Name and Value by NV Sequence

Fig. 8 shows a series of syllabus pages at "Anan National College of Technology"[2].

The page of type A in Fig. 8 (a) has 44 links. Three of them are the links to the top pages of the site, the college and the syllabi. The other 41 links are the links to course pages or pages of type B, e.g., "Applied mathematics" (Fig. 8 (b)), "Circuit theory", "Electromagnetics" and so on. These pages have the same template of 31 fields shown in Table 2. The second column "Freq." shows the frequency of the field text. For example, the 4th field text "course" appears in

[2] `http://www.anan-nct.ac.jp/gakka/syllabus/h13/curri_e.html`

```
procedure Pattern-and-Bs {
    Input: a listing page a;
    Output: tag-pattern;

     X = the list of pages linked from a;
     B = the maximal cluster in X w.r.t. base();
    D = d_1, ..., d_m = the maximal cluster in B w.r.t. tag();
    p = p_1.*.p_2.*....*.p_n.*.p_{n+1} the tag-sequence tag(d_1);
    return(p, B);
}

procedure ExtractFields {
    Input: a tag-pattern p = p_1.*.p_2.*....*.p_n.*.p_{n+1};
           an HTML file h;
    Output: a record a = (a[1], a[2], ..., a[n]);

    if tag(h)=p {
        a[i] = the i-th variable parts "*" in h;
    }
    return (a[1], a[2], ..., a[n]);
}

procedure NameValueSeparation {
    Input: a_1, ..., a_m : a list of records of the same fields,
        i.e., a_i = (a[i, 1], ..., a[i, n]);
        α : a threshold;
    Output: a list nv of "N" and "V" of length n;

    function g(i) = a[i,j] the j-th field of i-th record;
    for( j = 1; j <= n; j++ ) {
        weight[j] = the number of elements of maximal cluster w.r.t. g()
        in {1, 2, ···, m}
        if ( weight[j] > α) {
            nv[j] = "N";
        } else {
            nv[j] = "V";
        }
    }
    return nv;
}

main {
    Input: a keyword w;
    Output: a list of keywords;

    SearchResult = search("list of " + keyword);
    (p, B) = Pattern-and-Bs(SearchResult);
     C = map { λ(h) ExtractFields(p, h) } B;
    NV = NameValueSeparation(C);
    return the names of NV;
}
```

Fig. 5. Metadata Construction Algorithm.

41 pages, i.e., in all pages. A field text which appears in all pages is an attribute name or a common text to the site, such as a site name.

Field texts of Web documents in a series can be classified in two types.

N : Constant text that appears in almost all pages at the same position.
V : Variable text that varies page to page.

Maker	*Mitsubishi*
Model	*Lancer Evolution*
Year	*2003*
Mileage	*100*
Color	*Yellow*

(a) Vertical Alignment

Maker	Model	Year	Mileage	Color
Mitsubishi	*Lancer Evolution*	*2003*	*100*	*Yellow*

(b) Horizontal Alignment

Fig. 6. Name Value Alignment

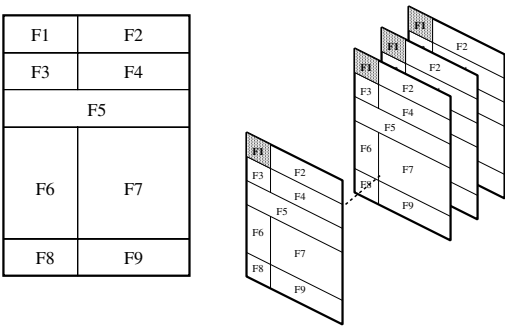
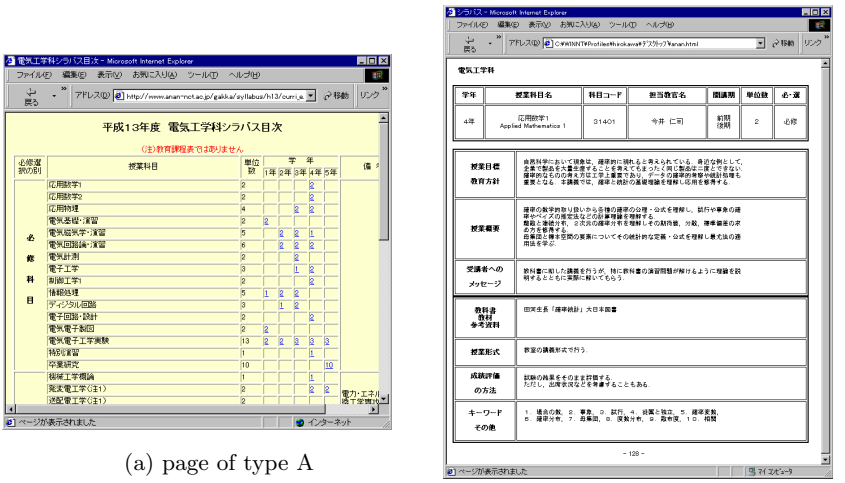


Fig. 7. Relative Position of Name and Value



(a) page of type A

(b) page of type B

Fig. 8. A Syllabus Page of Anan National College of Technology

"N" stands for name and "V" stands for value. Then we can describe the alignment of a page with some *NV sequences* N^nV^n. The i-th value is obtained at the position of i-th V.

Table 2. Appearance frequency of fields.

Field No.	Freq.	Word(s) (translated)	Word(s) (in Japanese)
1	41	Syllabus	シラバス
2	41	Dept. Elec. eng.	電気工学科
3	41	Grade	学年
4	41	Course	授業科目名
5	41	Code	科目コード
6	41	Lecturer Name	担当教官名
7	41	Period	開講期
8	41	Credit	単位数
9	41	elective or required	必・選
10	10	5th degree	5年
	15	4th degree	4年
14	28	First term, Second term	前期 後期
15	14	1	1
	24	2	2
16	17	elective (optional)	選択
	24	required (compulsory)	必修
17	41	Course Goal	授業目標 教育方針
19	41	Abstract	授業概要
21	41	Message to students	受講者へのメッセージ
23	41	Text books, Teaching Materials, Reference books	教科書 教材 参考資料
25	41	Class type	授業形式
27	41	Evaluation	成績評価 の方法
29	41	Keywords, MISC.	キーワード その他

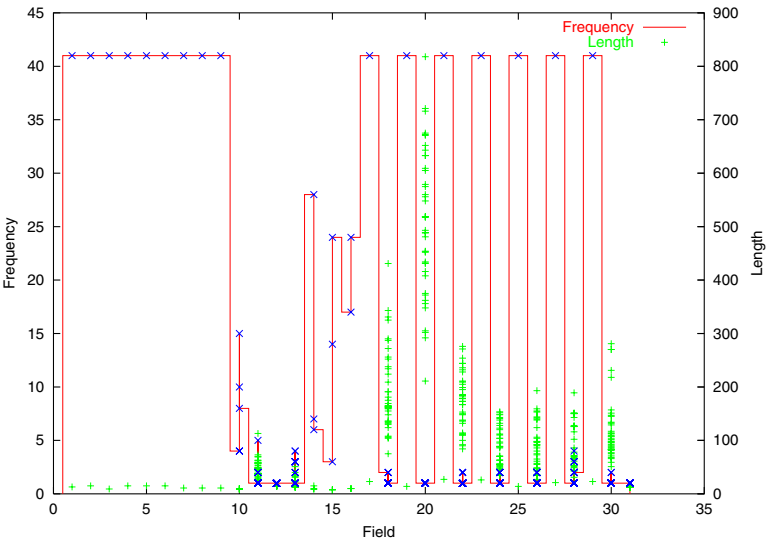


Fig. 9. Frequency and Length of Field Texts

Table 2 displays the names of high frequency. Note that another characteristic feature of value fields is that the length is relatively large.

Fig. 9 shows the frequency and the length of field texts. We can see that the text length is short and the frequency is high for the fields 1-9. On the other hand, the texts in the fields 10-16 are relatively long and have low frequency. If we see the fields 1-17 closely, we notice a discrepancy of the number of names (9) and the number of values (7). The first and second name fields are the cause

of the mismatch. There is no value corresponding to these "N". This kind of name can be considered as a record name instead of an attribute name. Thus we have the NV sequence $N^2N^7V^7(NV)^7$ and we can confirm that the record has 14 attributes.

6 Conclusion

We proposed a method for constructing metadata from a series of Web documents. It is often the case that a site provides many Web pages for a specific contents. Moreover these pages look very similar, because they are written with the same template HTML file. We used the common tag-sequence as the template of these pages. The individual contents of the pages are surrounded by these tags. The i-th string enclosed by the i-th and $i+1$-st tags can be considered as the i-th field of the record. If all of the i-th field are identical and can be considered as a constant, it does not represent the value but the name of an attribute in the record. The method we proposed uses the frequency of the string in the same field to distinguish the name and the value. The fields of names form a metadata for the series of Web documents.

Metadata construction is one of the key steps for integrating Web documents. The core idea of the method is to use the frequency of texts that appear in the same position of the similar semi-structured documents. The tag-sequence is used in the present paper, but there are many other similarity measurements that can be applied for detecting the similarity and for extracting a template of Web documents [5, 6, 7]. These approaches will improve the robustness of the proposed method.

References

[1] C. C. Aggarwal, F. Al-Garawi and P. S. Yu : "*Intelligent Crawling on the World Wide Web with Arbitrary Predicates*", Proc. WWW2001, 2001. `http://www10.org/cdrom/papers/110/index.html` 943

[2] A. Arasu and H. Garcia-Molina : "*Extracting Structured Data from Web Pages*," Proc. of ACM SIGMOD/PODS 2003 Conf., pp.337-348, 2003. 943, 945

[3] S. Chakrabarti, K Punera and M. Subramanyam : "*Accelerated Focused Crawling through Online Relevance Feedback*", Proc. WWW2002, 2002. `http://www2002.org/CDROM/refereed/336/index.html` 943

[4] C. H. Chang, S. C. Lui, Y. C. Wu : "*Applying Pattern Mining to Web Information Mining*," Proc. PAKDD 2001, Spring LNAI 2035, pp.4-16, 2001. 943, 944

[5] I. F. Cruz, S. Borisov, M. A. Marks and T. R. Webb : "*Measuring Structural Similarity Among Web Documents: Preliminary Results*", Proc. EP 1998, Springer LNCS 1375, pp.513–524, 1998. 944, 952

[6] S. Flesca, G. Manco, E. Masciari, L. Pontieri and A. Pugliese : "*Detecting Structural Similarities between XML Documents*", Proc. WEBDB2002, 2002. `http://feast.ucsd.edu/webdb2002/papers/19.pdf` 944, 952

[7] J. Han, J. Pei and Y. Yin : "*Mining Frequent Patterns without Candidate Generation*", Proc. ACM SIGMOD Intl. Conf. Management of Data, pp.1–12, 2000. 952

[8] J. W. Lee, K. Lee, W. Kim : "*Preparations for semantics-based XML mining*," Proc. IEEE Int. Conf. od Data Mining (ICDM) 2001, pp.345-352, 2001. 944

[9] K. Lerman, C. Knoblock and S. Minton : "*Automatic Data Extraction from Lists and Tables in Web Sources*", `http://www.cs.waikato.ac.nz/~ml/publications/1999/99SJC-GH-Innovative-apps.pdf` 943

[10] H. Leung, F. Chung, S. C. Chan : "*A New Sequential Mining Approach to XML Document Similarity Computation*," Proc. PAKDD 2003, Spring LNAI 2637, pp.356-362, 2003. 944

[11] T. Miyahara, Y. Suzuki, T. Shoudai, T. Uchida, S. Hirokawa, K. Takahashi and H. Ueda : "*Discovery of Frequent Tag Tree Patterns in Semistructured Web Documents*", Proc. PAKDD 2003, Springer LNAI 2637, pp.430–436, 2003. 944

[12] T. Taguchi, Y. Koga and S. Hirokawa : "*Integration of Search Sites of the World Wide Web*", Proc. CUM, Vol2, pp.25–32, 2000. 947

[13] S. Yamada, Y. Matsunaga, E. Itoh and S. Hirokawa : "*A study of design for intelligent web syllabus crawling agent*" Trans. of IEICE D-I, Vol.J86, No.8, pp.566-574, 2003. (in Japanese)

[14] 943, 944
M. Umehara, K. Iwanuma, H. Nagai :f "A Case-Based Semi-automatic Transformation from HTML Documents to XML Ones –Using the Similarity between HTML Documents Constituting a Series–," Journal of JSAI, Vol.16, No.5, pp.408-416, 2001. (in Japanese)

[15] C. C. Marshall : "*Making metadata: a study of metadata creation for a mixed physical-digital collection DL '98*," Proc. of the 3rd ACM Int'l Conf. on Digital libraries, pp.162-171, 1998.

[16] S. Handschuh and S. Staab : "*Authoring and annotation of web pages in CREAM*," Proc. WWW2002, 2002.
`http://www2002.org/CDROM/refereed/506/index.html` 945

[17] H. Stuckenschmidt and F. van Harmelen : "*Ontology-based metadata generation from semistructured information*," Proc. of K-CAP'01, pp.440–444, 2001 945