DBSJ Journal, Vol.13, No.1
March 2015

Applying XML Element Retrieval Techniques to Web Documents

Atsushi KEYAKI♡ Jun MIYAZAKI♡ Kenji HATANO♢

In this paper, we propose a method to expand XML element retrieval techniques into Web documents. XML element retrieval techniques return partial (sub) documents as search results, and are expected to be able to apply to other structured documents, namely, Web documents besides XML documents. The point is that physical document structures of Web documents are literally disorganized because Web documents are generated for not managing data but rendering on a Web browser. As another feature of Web documents, they contain many incomprehensive contents for human readers. To address challenges caused by these features, we propose 1) a reconstruction method of document structures according to logical structures of contents and 2) a filter for removing unimportant content which does not convey useful information to users. Our experimental evaluations showed that our proposed method improved search accuracy compared with both naive XML element retrieval approach and document retrieval approach.

1. Introduction

An XML element retrieval technique identifies descriptions which satisfy users' information need in an XML document. An element-based search system returns only the descriptions instead of the XML document, or alternatively, highlight the descriptions in the XML document. The effective way for returning a search result depends on applications. Since users can access information they need directly, an element-based search system saves users' time and energy in information retrieval.

In the past, XML element retrieval techniques were investigated with scientific articles and Wikipedia articles in the INEX project[1]. Consequently, an accurate XML element retrieval system is coming true. As a next step, since XML is not the only data format which is applicable with (XML) element retrieval, it is expected that these techniques are adapted into Web document which is also one of structured documents as same as XML document. A scope of XML element retrieval is spread drastically if an effectiveness for

♡ Member Graduate School of Information Science and Engineering, Tokyo Institute of Technology
keyaki@lsc.cs.titech.ac.jp, miyazaki@cs.titech.ac.jp
♢ Member Faculty of Culture and Information Science, Doshisha University
khatano@mail.doshisha.ac.jp

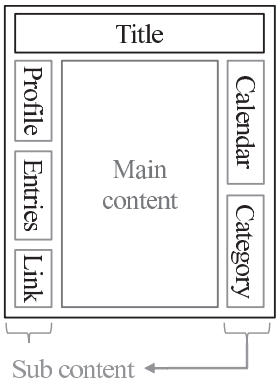

Figure 1: Typical structure of a Web document

```
<article>
  <p>Bill Gates is …</p>
  <body>
    <sec>Early life …</sec>
    <sec>Windows …</sec>
    <sec>Books …</sec>
  </body>
</article>
```

Figure 2: XML document

HTML documents is proved, because Web document is a very common data format.

In expanding XML element retrieval techniques into HTML document which is a representative data format of Web documents, characteristics of XML and HTML are quite different because of their intended purposes. We enumerate characteristics of HTML document as follows:

(1) many HTML documents are invalid in tag consistency,

(2) logical structure in terms of content and physical document structure do not agree with each other, and

(3) there are many parts which do not satisfy users' information need.

Concerning (1), consistency of start and end tags needs to be assured (every start tag corresponds to its end tag) when expanding XML element retrieval techniques into structured documents for treating elements properly. Generally speaking, most of XML documents are well-formatted in terms of tag consistency because they are mainly used for data management. In fact, many of XML parsers only accept XML documents of which tag consistencies are well-formatted. Conversely, the majority of HTML documents are not perfectly well-formatted in their tag consistency. This is caused by the fact that ordinary Web browsers interpret and display a HTML document even though tag consistency of the HTML document is deficient. To solve the problem, a tag balancer tool is often utilized. To cite an example, CyberNeko HTML Parser [2] is one of the most well used tag balancer tools which automatically complements the incomplete tags.

In regard to (2), many of hand written documents (even some of automatically generated documents) have logical document structure of content. For example, a document is composed of some chapters. Likewise, each chapter has some sections, and each section also has some sub-sections. This chapter composition information is definitely useful to understand content. XML element retrieval techniques exploit these information to identify an element which satisfies user's information need, because a document structure of an XML document is fundamentally based on a logical document structure of content. In other words, a document structure needs to be defined according to logical document structure of content in order to apply (XML) element retrieval techniques effectively.

In contrast, most of HTML tags perform text decoration, e.g., changes of font size, color, and style. In consequence,

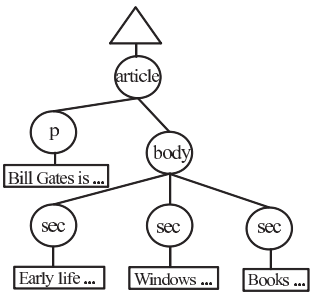

Figure 3: XML tree

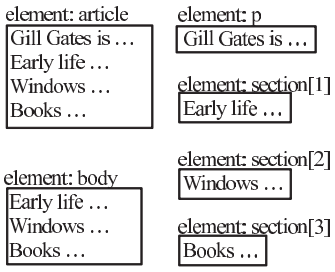

Figure 4: XML element

a document structure of HTML document does not reflected logical structure of content, which prevent expansion of XML element retrieval techniques into HTML documents.

Related to (3), a HTML document is often composed of multiple parts (elements). Concretely, a typical structure of Weblog articles is shown in Figure 1. It contains not only main–content but also further information such as a page title, a list of past entries, a list of link information, a category information of entries, and a set of tag cloud. Hereinafter, we call these further information as sub–content.

Generally, sub–content is just a hyperlink to other documents. Thus, users' information need is not satisfied only with these elements, although these are useful for exploratory information retrieval. The problem is that these elements tend to be ranked high in search results, because these elements where a unitary term often occurs frequently have a high relevancy score with a term-weighting-based scoring function. Therefore, we need to identify such an element composed of sub-content to be removed from search results. Note that we mainly focus on sub–content which harm search performance although there are some other sub–contents such as a profile information of the author, a calendar, and a search box.

To solve the problems raised by the characteristics of HTML documents, we format HTML documents; 1) reconstructing document structure according to logical structure of content, and 2) removing unimportant elements.

Concerned with 1), it is reported that some HTML tags represent a boundary between one topic to another [8]. We hypothesize that implicit logical document structure is estimated with these tags. Thus, we propose a method to reconstruct HTML documents using these HTML tags for achieving high agreement between a physical document structure and a logical document structure of content.

As a solution for 2), we propose a filter for removing unimportant elements, or sub–contents. These elements are expected to include many `A` tags because sub–content is a hyperlink to other documents. Therefore, we leverage occurrence information of `A` tags to eliminate sub–contents.

In the experiments described in Section 6, we survey search accuracy of the proposed method by comparison with a naive element retrieval method and a traditional document retrieval method to discuss about a potential of element-granule retrieval for HTML documents.

2. Overview of XML Element Retrieval

We show concrete examples in Figures 2, 3, and 4 to explain the definition of XML elements. Figure 2 illustrates an example of XML document. Figure 3 depicts a tree that is translated from Figure 2. An XML document can be expressed as a tree, which helps to understand the structure of the document.

The author of a document makes structures (e.g. chapters, sections, paragraphs.) We utilize these structures to identify the best description for satisfying the users' information needs. In other words, we suppose that texts in an item are based on the same theme. Then, we extract the item that discusses the theme of the users' interest.

A pair of start and end tags represents an XML element node in an XML tree, and the nested structure of XML elements represents an ancestor-descendant relationship. Each element in Figure 4 is the text that is composed of a set of text nodes in the XML tree in Figure 3. This demonstrates why there are overlapping XML elements in XML documents.

Suppose a user seeks information about "Early life …", "Windows …" and "Books …". XML element retrieval systems try to present an element whose root node is `body` to the user because the element contains all of the information that the user needs and no further information.

3. Related Studies

In article [12], it is reported that tags in structured documents are largely classified into two groups; A) tags surrounding self-contained content and B) tags enabling separate content. In this paper, we define tags of A) as structural tags. Concrete examples of structural tags are `HEAD`, `BODY`, and `P` tags of HTML. These tags are quite commonly used and can be meaningful clues for identifying useful and appropriate granular elements.

In addition, some HTML tags defined in HTML5 [3] such as `ARTICLE`, `SECTION`, `NAV`, and `ASIDE` tags are also structural tags. A `SECTION` tag can be nested, and each of the other tags represents specific or semantic context. This means that a physical document structure agreeing to a logical structure of the document can be generated with these tags. However, we propose a method which works well even when we cannot utilize these newly defined tags, because these tags do not widely used yet.

In contrast, tags of B) are used for representing decoration, attribute, and specific idea. To enumerate some examples, `B`, `FONT`, `I` tags are applicable. Most of HTML tags are classified into B), because HTML is defined for the sake of being used for displaying with a browser. The fact that the number of tags of A) is small suggests that it is more difficult to identify the most appropriate granular elements.

There are some kinds of tags which perform a boundary between one topic and another [8], for example, Heading tags (`H1`–`H6` tags), `HR` tag, and `BR` tag. These tags are leveraged to split content according to a topic.

A classification method separates HTML content based on rendered image [4]. A merit of the approach is that parts which are originally dissimilar in terms of document structure can be gathered into the same group if these parts are similar enough in regards to the appearance on a browser.

On the other hand, Yoshida et al. propose a method to extract main–content from news articles based on unsupervised learning [13]. The key idea of the study is that the blocks of sub–contents are the same or strongly similar to each other, while those of main–contents are different from each other.

4. Reconstruction of a Document Structure

In this section, we discuss a reconstruction method to solve the disagreement between a logical structure and a physical structure of a document.

A logical structure of a document is composed of a table of contents like chapters, sections, and paragraphs. In contrast, only the granularity of a paragraph can be defined with P in HTML documents. This causes a disagreement between a logical structure and a physical structure.

On the other hand it is reported that Heading tags (H1-H6) perform a boundary between one topic to another [8]. We suppose that a level of Heading tag (the number in a Heading tag) somewhat expresses a logical structure of content, because a level of Heading tag is set along with a degree of importance of content[1]. In short, we hypothesize that a more important Heading tag intends a larger granular topic while a less important Heading tag intends a smaller granular topic.

Information between a pair of start and end Heading tags (we call this as a Heading tag, for short) such as "bbb" in Figure 5 is just a title of a heading. It is general that contents just after a Heading tag such "ccc" and "ddd" in Figure 5 is about the title of the heading. We therefore reconstruct a HTML document according to a level of a Heading tag, because we expect that we can extract a logical structure of content with Heading tag information. Note that we insert a Container Heading tag for the goal.

- **Start tag:** When a Heading tag (Hx, $1 \leq x \leq 6$) appears, a start Container Heading tag (CHx) is inserted just before the Heading tag.
- **End tag:** When a level of a newly appeared Heading tag (Hx) is the same or smaller than that of previously appeared Heading tags, an end Container Heading tag (CHy, $1 \leq y \leq 6$) is inserted just before a start Container Heading tag (CHx). Note that the level of BODY is smaller than any Heading tag.

We show a concrete example of reconstruction process with Figure 5. The original HTML document in the left figure is reconstructed into that in the right figure. There are four Heading tags in the HTML document, e.g., H2, H3, H3, and H1 tags. When a H2 tag appears, a start CH2 tag is inserted just before the H2 tag. Next, a start CH3 tag is inserted just before the H3 tag, because the level of the H3 tag is larger than that of the previously appeared H2 tag. Then, a H3 tag appears again. Since the levels of the newly appeared H3 tag and the previously appeared H3 tag are the same, an end CH3 tag is inserted followed by a start CH3 tag. A H1 tag comes last. The level of the H1 is smaller than the previously appeared H2 and H3 tags. Accordingly, an end CH3 and an end CH2 tags are inserted followed by a start CH1 tag. The reconstruction process is completed when an end CH1 tag is inserted just before an end BODY tag.

Some other tags such as UL, OL, DL, TABLE, FORM, and DIV have possibilities that they are also regarded as structural tags. Likewise, BR and HR separate contents, which may be a clue of reconstruction. Managing these tags is a part of our future work.

[1]The more important a content is, the smaller the number is.

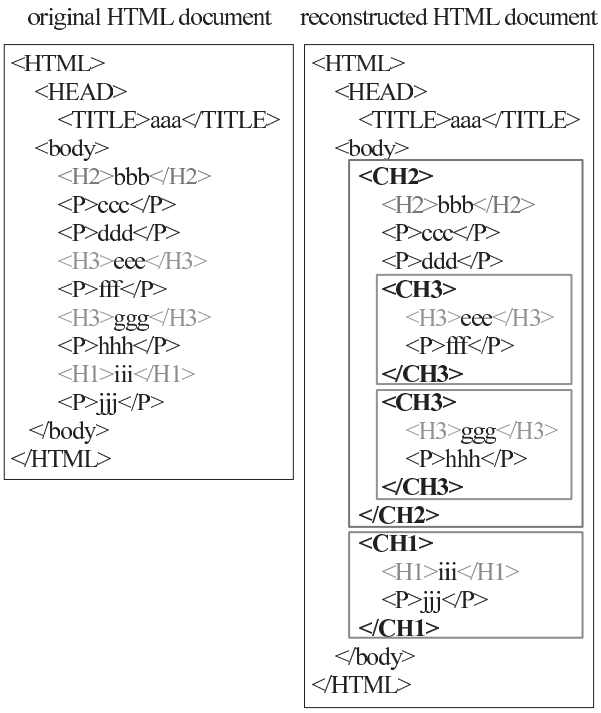

Figure 5: Reconstruction of an HTML document

5. Eliminating Elements of Outside of Main Body

Hereinafter, we describe how to remove unnecessary elements with a proposed filter. We regard elements composed of sub–content, or a list of past entries, a list of link information, a category information of entries, and a set of tag cloud, as these unnecessary elements because these elements does not satisfy users' information need directory.

Generally, elements of sub–content in the scientific articles and the Wikipedia articles are small in their length. That is the reason why elements with small length are filtered out with these articles [5]. However, elements of sub–content in HTML documents are not always short in their length.

Considering trends of elements composed of a list of entries or a list of link information, these are hyperlinks for transferring to other pages. Since these elements are expected to contain many A tags, we remove the elements which contain many A tags as sub–content. We enumerate candidate statistics for measuring a quantity of A tag.

- **A probability of occurrence of A tag:** A frequency of A tag normalized by element length.
- **A ratio of nodes rooted with A:** A ratio of the number of elements rooted with A tag to the total number of the elements in an XML tree.
- **A ratio of text size in A tag:** A ratio of the text size in A tags to the total text size of the element.

With regard to the first statistics, it is not always true that every element with high probability is sub–content. For example, such main–content that includes many figures linking to large-sized figures may be misjudged as sub–content. Thus, we conclude that this measurement is not the most appropriate one.

Concerned with the second statistics, elements containing many A tags are judged as sub–content. However, not every

element is marked up with other tags besides A tag. Since it is natural that main–content contain hyperlinks, the second statistics may also cause misjudgment.

In terms of the third statistics, it considers only A tags which are used for hyperlinks. Additionally, the statistics is independent of other tags besides A tag. It is expected that the statistics judge sub–contents properly. Therefore, we adopt the third statistics in our proposed filter.

Since the statistics measure a quantity of A tag, the elements of which quantity exceeds a threshold, τ, are removed as unimportant elements. We name this quantity as a sub–content score. Then, for element e, its sub-content score, $S_A(e)$, is calculated as follows:

$$S_A(e) = \frac{C_A}{C_e} \quad (1)$$

where C_A is the total number of characters in all descendant elements rooted with a A tag of e, and C_e is the total number of characters in e. The optimization of threshold $\tau(0 \leq \tau \leq 1)$ is discussed in the next section.

6. Experimental Evaluations

In this section, we first report the effect of the proposed reconstruction method and set the threshold value for the proposed sub–content filter, before evaluating search performance of the proposed method.

To confirm the effectiveness of XML element retrieval techniques for Web documents, we used Web (HTML) documents and Web queries for the experiments, i.e., 1CLICK-2 test collection [6]. It is composed of 1) a set of documents, 2) a set of queries, and 3) nuggets (ground truth) as a test collection. A set of queries is extracted from the query logs issued to a certain popular Web search system. Then, a set of documents contains top-ranked 500 results that the search system returns for each query. Moreover, nuggets are relevant texts for a query, which are provided by 1CLICK-2 organizers. We used all of them in the latter experiments unless otherwise specified.

It is common that documents for an evaluation of information retrieval systems are preprocessed before evaluations. The following preprocesses are applied before reconstructing the HTML documents.

1. removing attributes, comments, and special characters of HTML documents,
2. removing the stop words by SMART stop list [11],
3. applying stemming step by Porter [10], and
4. validating corresponding relations in tags with CyberNeko HTML Parser [2].

The PC that we used for the experiments runs Oracle Enterprise Linux 5.5. It has four Intel Xeon X7560 CPUs (2.3GHz), 512GB of memory, and a 4.5TB disk array. The indices were implemented using BerkeleyDB in GNU C++.

6.1 Evaluations of the Reconstruction Method

We investigated how document structures changed along with the reconstruction method. Table 1 shows the numbers and percentages of inserted Container Heading tags. These numbers correspond to the frequencies of Heading tags. As we can see from the table, H2 and H3 tags appeared frequently. Note that the average number of inserted Container Heading tag per document is 13.4.

Table 1: Inserted Container Heading tag

Tag name	the number of tags (%)
HC1	33,217 (.092)
HC2	113,300 (.31)
HC3	116,787 (.32)
HC4	60,276 (.17)
HC5	26,819 (.074)
HC6	10,880 (.030)

Table 2: Degree of deepening of document structures

degree of deepening	the number of documents (%)
0 (not deepened)	3639 (.14)
1	5194 (.19)
2	8300 (.31)
3	6912 (.26)
4	2509 (.093)
5	346 (.013)
6	28 (.0010)

Next, we examined the degree of deepening of document structure with Container Heading tags. To show some tangible examples, the degree of deepening of a document is 2 when the document includes H2 and H3 tags, while that is 6 when Heading tags appear in order from H1 to H6. We report the degrees of deepening of document structures and their numbers and ratios in Table 2. Heading tag is appeared in more than 80% of documents, and the average number of the degree of the deepening is 2.23.

It turned out that new granular elements apart from a paragraph are defined by focusing Heading tags. However, this also means an increase of search targets. Compared with the original documents, 25% of more elements are generated. We need to consider the way to control the explosion of newly generated elements when we target Web scale. It remains one of our future work.

6.2 Evaluation of Sub–content Filter

We explored whether the sub–content filter can remove sub–contents such as a list of past entries, a list of link information, a category information of entries, and a set of tag cloud, or not.

We randomly extracted 10 elements with each range of the sub–content score as Table 3 shows. We manually judged if each element is either main–content or sub–content. In consequence, every element of which sub–content score is higher than 0.7 is judged as sub–content. We therefore set 0.7 as threshold value τ for the sub–content score. Note that 9% of elements are removed as sub–content.

6.3 Effect of XML Element Retrieval Techniques for HTML Documents

6.3.1 Experimental Design

There are eight kinds of query types in 1CLICK-2 queries. Above them, we used only 15 DEFINITION type queries. The aim of this query type is to extract relevant information exhaustively. Thus, we suppose that this query type is relatively compatible with element retrieval.

Table 3: Discriminant accuracy of the sub–content filter

sub–content score	accuracy
0.1-0.3	50%
0.3-0.5	60%
0.5-0.7	80%
0.7-0.9	100%
0.9-	100%

Table 4: Agreement rate, exhaustiveness rate, and F–measure

	@1			@5			@10		
	AR	ER	FM	AR	ER	FM	AR	ER	FM
ELEM	.503	.233	.176	.498	.305	.298	.512	.351	.344
REC	.517	.282	.230	.545	.372	.424	.586	.360	.418
DOC	.382	.800	.458	.372	.733	.413	.394	.727	.504

Table 5: Average text sizes and standard deviations at top-10

	average text size	standard variation
ELEM	2288.754	2576.560
REC	1717.870	1292.166
DOC	6411.987	1486.383

We evaluated search performances at three levels, namely, top-1, top-5, and top-10 in Table 4. There are three methods to compare, i.e., element search approach (ELEM), reconstruction approach (REC), document search approach (DOC). In generating search results of ELEM, we firstly calculate a relevancy score of each element with BM25E [9] which is one of the most popular term weighting schemes for element retrieval. Note that we adopt the tag-based approach for global weight calculation, because the number of documents is not enough for calculating accurate global weights with the path expression-based approach. To generate search results of REC, we additionally apply the reconstruction method proposed in our previous work [7] when an overlap occurs. This method chooses the best granular elements as search results from overlapped elements. We omit a detailed explanation of the method because of space limitation. Note that both ELEM and REC adopt the sub–content filter and eliminate elements shorter than 15 in their length. Search results of DOC contain elements rooted with a whole document only.

Next, we explain the evaluation method. We manually extract relevant descriptions from documents that search results belong to. During the process, we did not assign the importance of descriptions but evaluate whether the descriptions are relevant or not.

The evaluation measures that we used are agreement rate (AR), exhaustiveness rate (ER), F–measure of AR and ER (FM), average text size, and standard deviation. Let e be a retrieved element as search result and D_e be a document that e belongs to. The values of the measures are calculated as follows:

$$AR = \frac{size_{e,r}}{size_e} \tag{2}$$

$$ER = \frac{size_{e,r}}{size_{D_e,r}} \tag{3}$$

$$FM = \frac{2 \cdot AR \cdot ER}{AR + ER} \tag{4}$$

where $size_{e,r}$ is the text size of relevant descriptions in e, $size_e$ is the text size of e, and $size_{D_e,r}$ is the text size of relevant descriptions in D_e. Note that these are average values calculated by using 15 queries.

6.3.2 Results of the Experiments

The results tell that the agreement rates of both ELEM and REC are higher than those of DOC at all levels. On the other hand, the exhaustiveness rate of DOC is higher than those of ELEM and REC. Meanwhile, largely F–measure of DOC is the highest except that F–measure of REC at top-5 is higher than that of DOC. In consequence, the element retrieval approach brings higher agreement rates.

Moreover, both agreement rate and exhaustiveness rate improved at all levels with the reconstruction method. As a result of multiple comparisons on agreement rates, there is a statistical difference between REC and ELEM at significance level 5%, while there are statistical differences between both REC and DOC, ELEM and DOC at significance level 1%.

In terms of average text sizes and standard deviations at top-10 as we show in Table 5, the average text size of DOC is much larger than those of ELEM and REC. Entire documents are returned as search results with ELEM when appropriate granular elements cannot be returned. Thus, the average text size and the standard deviation of ELEM is higher than those of REC. As a result, REC is more helpful for users, because a smaller amount of retrieved search requires less laborious.

6.3.3 Positive and Negative Examples of the Reconstruction Method

As a result of experiments described in the previous section, the reconstruction method largely improved search accuracy. Figure 6 depicts the example of positive effect of the reconstruction method. Newly defined elements, i.e., `CH2` and `CH3`, are composed of multiple paragraphs. With these elements, the proposed system is able to return middle granular elements.

In more detail, as shown in Figure 8 which depicts the simplified source code of the document, the document has a flat structure. To extract descriptions satisfying users' information need, each sentence denoted as `P` tag is not useful. On the other hand, Figure 9 shows the simplified source code of the reconstructed document. Newly defined granular elements, namely `CH2` and `CH3`, whose sizes are larger than a paragraph and smaller than an entire document, can be returned as a search result.

Meanwhile, the reconstruction method causes harmful effect in some cases as Figure 7 shows. The element generated by the Heading tag appearing last in a document may include sub–contents and footer information. This decreases search performance. To avoid this, we need to regulate these unimportant parts as a part of our future work.

Moreover, we observed one more problem that we cannot remove some sub–contents with the sub–content filter from reconstructed documents. It seems that the appropriate threshold value for the sub–content filter changes by granularity of an element. Precisely, even elements composed of sub-contents may have a lower sub–content score for larger granular elements. There are some candidate approaches for resolving the problem, for example, smoothing with element length, utilizing rendered visual information, and machine learning for discriminating sub–contents.

7. Conclusions

In this paper, we adapted XML element retrieval techniques to HTML documents to return accurate and focused search results compared with traditional document retrieval. In adapting the techniques, there are three kinds of features of HTML documents compared with XML documents; (1) the document structures of HTML documents are not well-formatted, (2) there is disagreement between the logical structure of document content and physical document

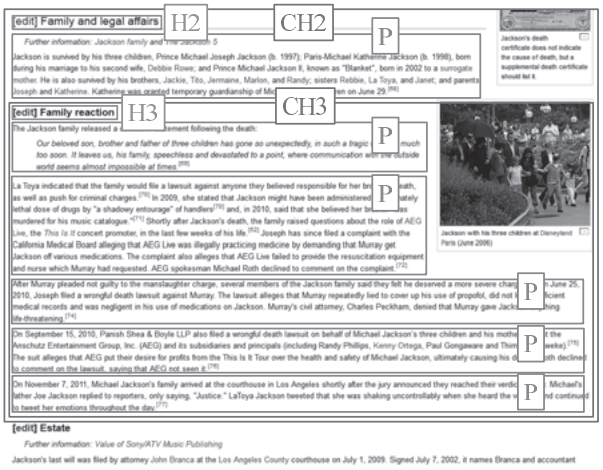

Figure 6: Multiple paragraphs are aggregated into one element

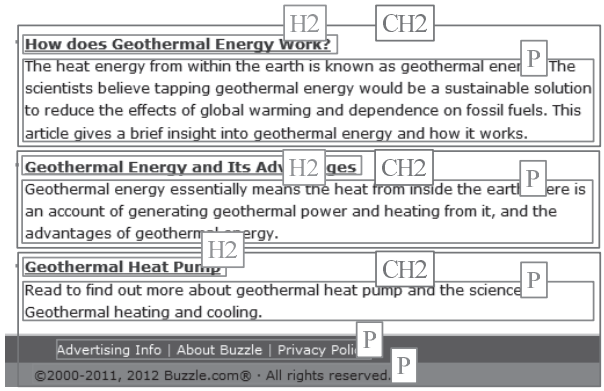

Figure 7: Newly generated element including sub–content

```
<H2>Family and legal affairs</H2>
<P>Further information: Jackson family and The Jackson 5<P>
<P>Jackson is survived ... Michael's three children on June 29.<P>
<H3>Family reaction</H3>
<P>The Jackson family released ... following the death:</P>
<P>Our beloved son, ... almost impossible at times.</P>
<P>La Toya indicated that ... comment on the complaint.</P>
<P>After Murray pleaded ... anything life-threatening.</P>
<P>On September 15, 2010, ... saying that AEG not seen it.</P>
<P>On November 7, 2011, ... her emotions throughout the day.</P>
```

Figure 8: Simplified original HTML document

```
<CH2>
  <H2>Family and legal affairs</H2>
  <P>Further information: Jackson family and The Jackson 5<P>
  <P>Jackson is survived ... Michael's three children on June 29.<P>
  <CH3>
    <H3>Family reaction</H3>
    <P>The Jackson family released ... following the death:</P>
    <P>Our beloved son, ... almost impossible at times.</P>
    <P>La Toya indicated that ... comment on the complaint.</P>
    <P>After Murray pleaded ... anything life-threatening.</P>
    <P>On September 15, 2010, ... saying that AEG not seen it.</P>
    <P>On November 7, 2011, ... her emotions throughout the day.</P>
  </CH3>
</CH2>
```

Figure 9: Simplified reconstructed document

structure, and (3) HTML documents contain many uninformative contents. These features may prevent adapting XML element retrieval techniques to HTML documents. Since the first feature is solved with an existing tool, we proposed a method to reconstruct HTML documents hinted by some tags which represent boundary of contents for diminishing the disagreement, and proposed a filter for eliminating unimportant elements with the ratio of the hyperlinks of an element.

As a result of experimental evaluations, the filter properly removes unimportant elements, while the reconstruction method improved search performance with decreasing text size of search results. Moreover, it is revealed that the framework of element retrieval can return more focused and accurate search results compared with those of document retrieval. On the other hand, the reconstruction method may generate elements including uninformative descriptions, which cause a drop in search performance.

Leveraging structural tags besides Heading tags for reconstruction of a document and refining the sub–content filter are parts of our future work.

[Acknowledgments]

This work was partly supported by JSPS KAKENHI Grant #26280115 and #25540150.

[Bibliography]

[1] Initiative for the evaluation of xml retrieval. http://inex.mmci.uni-saarland.de/. Accessed in November, 2013.

[2] Marc Guillemot Andy Clark. Cyberneko html parser (1.9.19). http://nekohtml.sourceforge.net/index.html, 2013. Accessed in November, 2013.

[3] Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Edward O'Connor, Silvia Pfeiffer, and Ian Hickson. HTML5. http://www.w3.org/TR/html5/, 2014. Accessed in November, 2013.

[4] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Extracting Content Structure for Web Pages based on Visual Representation. In *Proc. of the 5th ACM APWeb*, 2003.

[5] Kenji Hatano, Hiroko Kinutani, Toshiyuki Amagasa, Yasuhiro Mori, Masatoshi Yoshikawa, and Shunsuke Uemura. Analyzing the Properties of XML Fragments Decomposed from the INEX Document Collection . In *Advances in XML Information Retrieval*, volume 3493 of *LNCS*, pages 168–182. Springer Berlin, 2005.

[6] Makoto P. Kato, Matthew Ekstrand-Abueg, Virgil Pavlu, Tetsuya Sakai, Takehiro Yamamoto, and Mayu Iwata. Overview of the NTCIR-10 1CLICK-2 Task. In *Proc. of the 10th NTCIR Conference*, 2013.

[7] Atsushi Keyaki, Kenji Hatano, and Jun Miyazaki. Result Reconstruction Approach for More Effective XML Element Search'. *International Journal of Web Information Systems (IJWIS)*, 7(4):360–380, 2011.

[8] Seung-Jin Lim and Yiu-Kai Ng. Converting the Syntactic Structures of Hierarchical Data to Their Semantic Structures. *Information Organization and Databases*, 579:343–355, 2000.

[9] Wei Liu, Stephen Robertson, and Andrew Macfarlane. Field-Weighted XML Retrieval Based on BM25. In *Formal Proc. of INEX 2005 Workshop*, volume 3977 of *LNCS*, 2006.

[10] M.F.Porter. An Algorithm for Suffix Stripping. In *Computer Laboratory, Cambridge*, 1980.

[11] G. Salton. *The SMART Retrieval System–Experiments in Automatic Document Processing*. Prentice-Hall, 1971.

[12] Takashi Tokuda and Keishi Tajima. Classification of XML Tags according to Their Roles in Document Structure. *DBSJ Journal*, 8(1):1–6, 2009. (in Japanese).

[13] Mitsuo Yoshida and Makio Yamamoto. Primary Content Extraction from News Pages without Training Data. *DBSJ Journal*, 8(1):29–34, 2009. (in Japanese).

Atsushi KEYAKI

He is an assistant professor at the Graduate School of Information Science and Engineering, Tokyo Institute of Technology. He received the Ph.D. degree in engineering from the Nara Institute of Science and Technology (NAIST) in 2014. He was a JSPS research fellow (DC2) from 2012 to 2014, and a research intern at Microsoft Research Asia in 2013.

Jun MIYAZAKI

He joined the faculty of Department of Computer Science, Tokyo Institute of Technology in 2013, where he is currently a professor. Before joining Tokyo Tech., he served as an associate professor at Nara Institute of Science and Technology (NAIST). He is a member of ACM, IEEE Computer Society, IEICE, IPSJ, and HIS.

Kenji HATANO

He is currently an associate professor of Faculty of Culture and Information Science, Doshisha University. Before joining Doshisha University, he served as a postdoctoral fellow at Kobe University and an assistant professor at Nara Institute of Science and Technology. He is a member of ACM, IEEE Computer Society, IEICE, and IPSJ.