IEICE TRANS. INF. & SYST., VOL.E89–D, NO.6 JUNE 2006

PAPER Special Section on Human Communication II

# Extraction of Semantic Text Portion Related to Anchor Link

**Bui Quang HUNG**[†a)], **Masanori OTSUBO**[†], *Nonmembers*, **Yoshinori HIJIKATA**[†], *Member*, *and* **Shogo NISHIDA**[†], *Fellow*

**SUMMARY** Recently, semantic text portion (STP) is getting popular in the field of Web mining. STP is a text portion in the original page which is semantically related to the anchor pointing to the target page. STPs may include the facts and the people's opinions about the target pages. STPs can be used for various upper-level applications such as automatic summarization and document categorization. In this paper, we concentrate on extracting STPs. We conduct a survey of STP to see the positions of STPs in original pages and find out HTML tags which can divide STPs from the other text portions in original pages. We then develop a method for extracting STPs based on the result of the survey. The experimental results show that our method achieves high performance.

***key words:*** *text mining, web mining, semantic text portion, link structure, anchor, user experiment*

## 1. Introduction

In the field of Web mining, many researchers come to focus on the link structure. When there is a link from a web page to another one, the former is called the **original page** and the latter is called the **target page**. One target page may have many original pages. One of the most important characteristics of the link structure is that the text portions around the anchors in the original pages describe the target pages [1]. Henzinger, in his survey on the link structure analysis [2], explains that this characteristic originates from the following human factor. Many authors of original pages create links because they think the links are useful for the readers. A link from an original page to a target page can be seemed as a recommendation about the target page by the author of the original page. The author also writes some texts around the anchor to explain the target page to the readers from his own viewpoint. These text portions are semantically related to the target page. We give the following definition about this kind of text portions.

**Definition 1** Semantic text portion (STP) in an original page is a text portion which is semantically related to the anchor pointing to the target page.

Recently, STP is getting popular in the field of Web Mining. STPs can be used for many applications. One example is automatic summarization ([3]–[5]). STPs may include important information about target pages. We can make summaries of target pages by collecting them. Another example is document categorization ([6]–[11]). Because the target page contains many noise parts such as banner ads and links for navigation, STPs may represent the content of the target page better. Compared to using the text of the target page, there is a possibility that we can make a better directory by using STPs.

Researchers have proposed various methods for extracting STPs. These methods are anchor-text method, fixed-window method, sentence-based method, paragraph-based method, and list-based method. The anchor-text method is the simplest one. It extracts the text portion between the tags <A> and </A> of the anchor. The fixed-window method extracts the anchor text and the predetermined number of words around the anchor. The sentence-based method extracts one or more sentences around the anchor. The paragraph-based method extracts the paragraph which begins with the anchor followed by texts. The list-based method extracts the list item which includes the anchor. The details of these methods are explained in Sect. 3.

These methods are too simple to extract all the STPs in one original page. The problem of extracting STPs is that they locate in various kinds of location like the text around the anchor, the page title, the list title, the first row of the table and so on (Examples are shown in Sect. 2). Therefore the previous methods cannot extract STPs in high precision and especially in high recall.

Our approach to solve this problem is as follows. We conduct a survey of STPs to see which kinds of text portion in an original page are related to the anchor. We hope that we will find out some HTML tags which can semantically divide STPs from the other text portions in original pages. Based on the result of the survey, we develop a method for extracting STPs. Our method represents an original page by a DOM tree to analyze its document structure. DOM (Document Object Model) is an API to access any parts of a Web page which is standardized by W3C [12]. Our method then extracts STPs by using specific HTML tags which are found in the survey.

The most serious shortcoming in the previous researches is that they did not survey where STPs are written in an original page and did not evaluate their methods from the viewpoint of extracting STPs. They only proposed their simple methods and used the text portions extracted by their methods for upper-level applications. They did not consider

Manuscript received September 27, 2005.
Manuscript revised January 10, 2006.

[†]The authors are with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560–8531 Japan.
a) E-mail: bqhung@nishilab.sys.es.osaka-u.ac.jp

DOI: 10.1093/ietisy/e89–d.6.1834

Copyright © 2006 The Institute of Electronics, Information and Communication Engineers

whether the extracted text portion itself is semantically related to the target page or not.

In our research, we conducted a deep survey of the location of STPs and evaluated our method from the viewpoint of extracting STPs by inviting three evaluators. We made a dataset which consists of more than 1000 real original pages for the survey and a dataset which consists of 200 real original pages for the evaluation. The evaluators judged which text portions are real STPs in those pages. We decided on the text part which is a real STP by the majority vote. In the evaluation, we compared the texts extracted by our method to the real STPs given by evaluators. We then compared our method to the previous methods in extracting STPs. The experimental results showed that our method can achieve high precision and also the highest recall among the previous methods.

Even if we know that our method extracts STPs in higher precision and in higher recall than other methods, we do not know how much difference the upper-level application produces in actual users' usages. Finally we applied our method to summarization as an upper-level application. We summarized the original pages to one target page. We compared the summaries created by our method to the summaries created by the most popular existing method in the user experiment where users should work on a specific task.

In brief, the contributions of this paper are as follows:

- We deeply survey the locations of STP in original pages for the first time.
- We propose a method for extracting STPs from the result of the survey.
- We evaluate extracted text portions by using real STPs given by evaluators for the first time.
- We see the user's performance for completing tasks when texts extracted by our method are applied to an upper-level application.

The rest of this paper is organized as follows. In Sect. 2, we give some examples of STPs. In Sect. 3, we discuss the related works for our research. Section 4 discusses the survey of STP and Sect. 5 explains our method for extracting STPs. In Sect. 6, we evaluate STPs extracted by our method and compare our method to other methods. In Sect. 7, we apply our method to summarization and see the effectiveness in the user experiment. Section 8 provides some concluding remarks and directions for future research.

## 2. Examples of STPs

We give some examples of STPs to show the importance of its content and the variety of its location. STPs may include facts or people's opinions (evaluation and categorization) about target pages. A fact is information about what content is written in the target page or what service is offered in the target page. Evaluation is information about how people think the target page. Categorization is information about which category people assign the target page to. We provide three examples in Fig. 1 to Fig. 3.

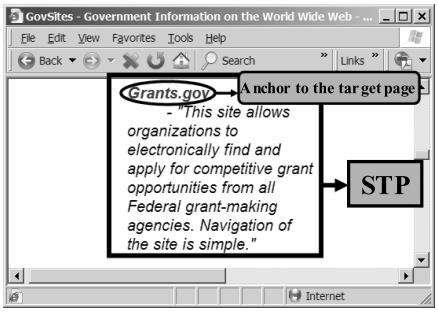



**Fig. 1** The STP is around the anchor and includes a fact and the people's evaluation about the target page.

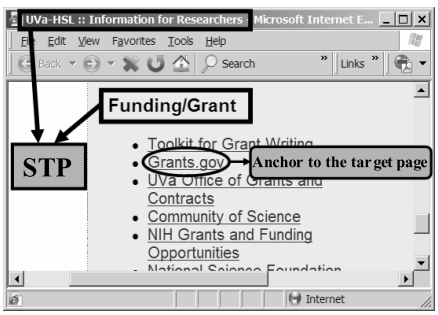



**Fig. 2** The STPs are the page title and the list title. One of them is a fact and the other is the people's categorization about the target page.

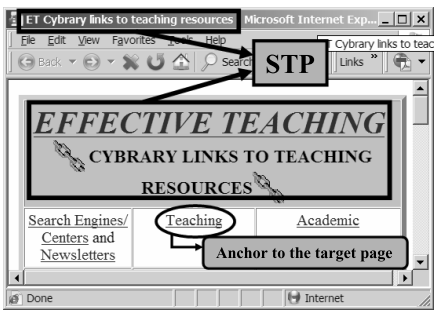



**Fig. 3** The STPs are the page title and the first row of a table. The STPs include a fact and the people's evaluation about the target page.

In Fig. 1, the STP is the paragraph after the anchor. Its content is *"This site allows organizations to electronically find and apply for competitive grant opportunities from all federal grant-making agencies. Navigation of the site is simple"*. We can consider the first sentence as a fact about the target page because this site introduces many competitive grants and offers electronic forms to apply for them. The second sentence is the author's evaluation about the target page because other users may think that the navigation of this site is complicated.

In Fig. 2, when the second anchor ("Grants.gov") is the anchor to the target page, the STPs are the page title *"Information for Researchers"* and the list title *"Funding/Grant"*. When we saw the actual Web site of this target page, we found that they offer not only information about grants but also information about contracts with companies. Therefore

the page title represents the fact about the target page. Because we found that funding or grant is one of the services in the target page, we can see that it is the author of the original page who categorizes the target page into "Funding/Grant" category. It is the categorization about the target page.

In Fig. 3, when the first anchor ("Teaching") is the anchor to the target page, the STPs are the page title *"ET cybrary links to teaching sources"* and the first row of the table *"EFFECTIVE TEACHING"*. The former is the fact about the target page because we found that the target page gives a teaching resource when we checked the target page. The latter is people's evaluation because the author of the original page thinks that the teaching resource is effective, but others may not.

From these examples, we can see that STPs contain important information of the target page and also written in various places.

## 3. Related Works

This section introduces researches on extracting STPs and researches on using STPs for the upper-level application as related works.

Recently, many researchers in the field of Web mining use the link structure for various purposes such as automatic summarization, document categorization, and page-ranking ([1], [3]–[11], [13]–[15]). Many researchers not only use the graph structures of links, but also exploit STPs ([1], [3]–[11], [13]). Some of them exploit STPs for summarizing web pages ([3]–[5]). Some of them use STPs for categorizing web pages ([6]–[11]). The others exploit STPs for ranking web pages ([1], [13]).

To extract STPs, previous researchers propose various kinds of method. The simplest method is the **anchor-text method** ([1], [4], [11]). This method extracts the text portion between the tags <A> and </A> of the anchor. Davison's work shows that anchor texts are related to contents of target pages but they do not contain enough information about target pages [1]. In many cases, they are only the URLs of the target pages. In order to get more information about target pages, some researchers also use the following two methods for extracting STPs: the **fixed-window method** ([8], [13]) and the **sentence-based method** [3]. The fixed-window method extracts the anchor text and the predetermined number of words around the anchor. The size of the fixed-window is important. If it is small, the extracted text portion may not be enough. If it is large, the extracted text portion may include many noise keywords. Both works [8], [13] use 50 words before and after the anchor. The sentence-based method extracts one or more sentence(s) around the anchor. Clearly, the number of sentences is important. The **paragraph-based method** extracts the paragraph which begins with the anchor followed by texts ([5], [10]). Using only this pattern is not enough because of the following two reasons. The first one is that the anchor is not only at the beginning of a paragraph. The second one is that the anchor exists not only in the paragraph but also in other objects in an original page such as table and list. The **list-based method** extracts the list item which directly includes the anchor [9].

These methods only consider the text portion near the anchor. Other researches use text portions which exist far from the anchor ([6], [9], [10]). **Furnkranz's method** and **Attardi's method** extract the page title and all the headers (H1 to H6) of the original pages ([9], [10]). **Roy's method** also extracts the page title and specific type of headers [6]. If there are several headers at the same level, it extracts the nearest one to the anchor. It also extracts the nearest decorated text portion to the anchor like strong, bold, italicized, or emphasized if there is no header between this text portion and the anchor. However these methods cannot extract STPs in other objects like list and table.

## 4. Survey of STP

In this section, we explain our survey of STP. The purpose of this survey is to see the positions of STPs in original pages and find HTML tags which can divide STPs from the other text portions in original pages. We realize that there are two types of STP from the viewpoint of its locations. One type exists around the anchor. This means that it directly includes the anchor (see Fig. 1). The other type exists in the upper-level structure of the original page. A web page is described in HTML and all parts of the web page (document) are structured by tags. The latter type does not touch the anchor and exits in the upper-level of this document structure (see Fig. 2 and Fig. 3). We call the former type the **Local Semantic Portion (LSP)**. We call the latter type the **Upper-level Semantic Portion (USP)**. Our survey consists of the survey of LSP and the survey of USP.

### 4.1 Dataset and Survey Method

We prepared 1108 real original pages in our survey. These 1108 web pages are 752 original pages of 50 official target pages such as a government's web page and a company's web page and 356 original pages of 50 personal target pages such as an individual's web page about his hobby. We collected these original pages as follows. We randomly selected 50 official target pages and 50 personal target pages from Open Directory [16]. For each target page, we found its original pages by using Google [17]. To get original pages of a target page, Google offers a search function by the query type *"link:URL of the target page"*. We used 20 original pages at most for each target page.

We invited three evaluators to give us the right answer of STPs. The method we used in the survey is as follows. For each original page in the dataset, we show the three evaluators its content and the anchor pointing to its target page. The evaluators see the content of the target page. After that, we ask them to judge which text portions are semantically related to the anchor. We define a real STP as the text portion which is judged to be semantically related to the anchor by at least two evaluators. The detail of how to define a

real STP is as follows. Let i shows IDs of original pages (i =1...1108), and A, B and C show IDs of three evaluators. $P_{iA}$, $P_{iB}$, and $P_{iC}$ shows the STPs extracted from the *i-th* original page by three evaluators. We judge whether the text should be a real STP or not by word. We count the number of evaluators who include the word. If more than two evaluators include the word in their STPs, the word becomes a real STP (also see Fig. 4). We call the real STP in the *i-th* original page $P_i$.

## 4.2 Survey of LSP

### 4.2.1 Positions of LSPs in Original Pages

Through the survey, we realized that LSPs are located in one of the following five places: table, list (ordered and un-ordered list), definition list, paragraph, or DIV object. Table 1 shows the number of LSPs in each place in 1108 original pages.

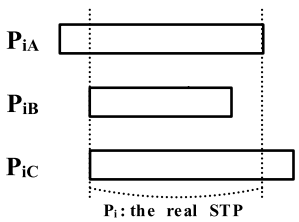



**Fig. 4** The real STP in the i-th original page

**Table 1** Number of LSPs in each place.

| Position | Total |
|---|---|
| Paragraph | 320 |
| Ordered and un-ordered list | 354 |
| Definition list | 56 |
| Table | 339 |
| DIV | 39 |

### 4.2.2 HTML Tags for Dividing LSPs from the Other Text Portions

This subsection explains the result of survey about what kind of HTML tag can divide the LSP from the other text portions in each place.

**a) When the LSP is in a paragraph:**

We found that there are the following two cases when a LSP is in a paragraph. After here, we call the paragraph which has the anchor to the target page the current paragraph.

- **Case 1:** The LSP covers the whole paragraph.
  In this case, we realized that there is only one anchor in the paragraph or there is no <BR> tag, which is a tag to give a line feeder, in the paragraph. We found that <P> tag and </P> tag can divide the LSP from the other text portions.
- **Case 2:** The LSP is one part of the paragraph.
  In this case, we realized that there are several anchors and several <BR> tags in the paragraph. We also found that there are four sub-cases as shown in Fig. 5.
  - **Sub-case 1:** The paragraph begins with an anchor followed by texts and there is no <BR> tag between the anchor and the following texts.
  - **Sub-case 2:** The paragraph begins with an anchor followed by texts and there are one or more <BR> tag(s) between the anchor and the following texts.
  - **Sub-case 3:** The paragraph begins with texts and there is no <BR> tag between the texts and the following anchor.
  - **Sub-case 4:** The paragraph begins with texts and there are one or more <BR> tag(s) between the texts and the following anchor.

  We found that in Sub-case 1 and Sub-case 2, the LSP is divided by the <BR> tag before the anchor and the <BR> tag before the next anchor. We also found that

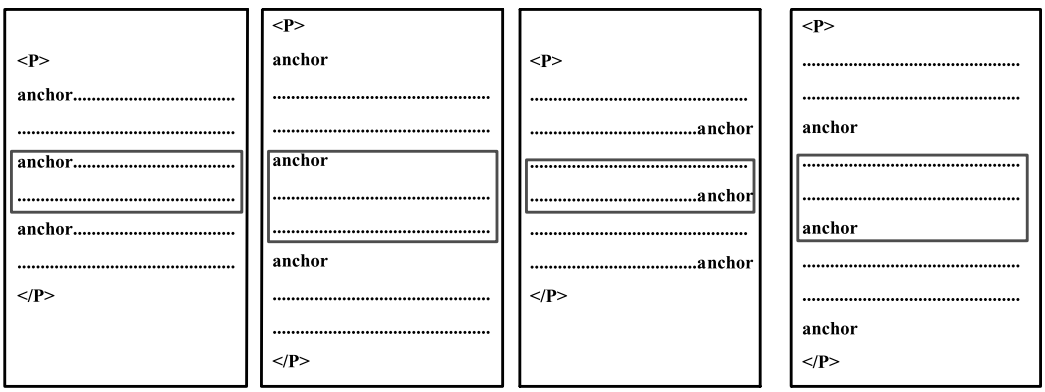


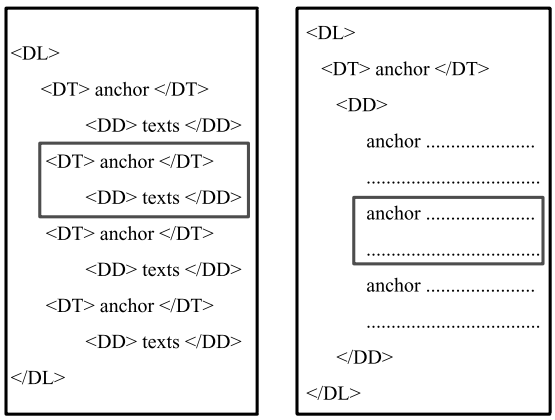

**Fig. 5** Four cases when the LSP is one part of a paragraph.

in Sub-case 3 and Sub-case 4, the LSP is divided by the <BR> tag after the previous anchor and the <BR> tag after the anchor.

**b) When the LSP is in a list:**
We found the following two cases. After here we call the list item which includes the anchor to the target page the **current list item**. We call the list which has the current list item the **current list**.

- **Case 1:** The LSP covers the whole current list item.
  We found that the LSP is divided from the other text portions by the <LI> tag and </LI> tag.
- **Case 2:** The LSP is one part of the current list item.
  We found that the LSP is divided from the other text portions by <BR> tags like a)-Case 2.

**c) When the LSP is in a definition list:**
We found the following two cases (also see Fig. 6).




i) The LSP covers the definition term including the anchor and the definition description.

ii) The LSP is a part of the definition description and there are several anchors and several line feeder tags in the definition description.

**Fig. 6** Two cases in which the LSP is in a definition list.

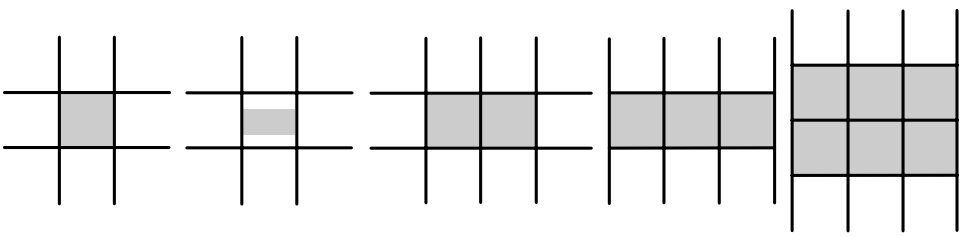
- **Case 1:** The LSP covers the definition term including the anchor and the definition description of the definition term.
  We found that the LSP is divided from the other text portions by the <DT> tag before the anchor and the </DD> tag after the anchor.
- **Case 2:** The LSP is one part of the definition description.
  We found that there are several anchors and several <BR> tags in the definition description. The <BR> tag before the anchor and the <BR> tag before the next anchor can divide the LSP from the other text portions.

**d) When the LSP is in a DIV object:**
We found the following two cases.

- **Case 1:**The LSP covers the whole DIV object.
  We found that the LSP is divided from the other text portions by the <DIV> tag and </DIV> tag.
- **Case 2:** The LSP is one part of the DIV object.
  We found that the LSP is divided from the other text portions by <BR> tags like a)-Case 2.

**e) When the LSP is in a table:**
We found that there are the following five cases (also see Fig. 7). After here we call the cell where the anchor to the target page exists the **current cell**. We call the row which has the current cell the **current row**. We call the table which has the current row the **current table**.

- **Case 1:** The LSP covers the whole current cell.
  We found that the LSP is divided by <TD> tag and </TD> tag.
- **Case 2:** The LSP is one part of the current cell.
  We found that the LSP is divided from the other text portions by <BR> tags like a)-Case 2.
- **Case 3:** The LSP covers several cells (not all cells) in the current row.
  We found there are the following two sub-cases (also



i) The LSP covers the current cell

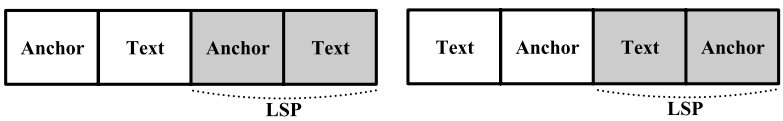
ii) The LSP is one part of the current cell

iii) The LSP covers several cells of the current row

iv) The LSP covers the current row

v) The LSP covers several rows

**Fig. 7** Five cases in which the LSP is in a table.




i) The row begins with an anchor

ii) The row begins with texts

**Fig. 8** Two sub-cases in which the LSP covers several cells of the current row.

see Fig. 8).

- **Sub-case 1:** The current row begins with an anchor.
- **Sub-case 2:** The current row begins with texts.

We found that in Sub-case 1, the <TD> tag before the anchor and the </TD> tag before the next anchor divide the LSP from the other text portions. We found that in Sub-case 2, the <TD> tag after the previous anchor and the </TD> tag after the anchor can divide the LSP from the other text portions.

- **Case 4:** The LSP covers the current row.
We found that the <TR> tag and </TR> tag of the current row can divide the LSP from the other text portions.
- **Case 5:** The LSP covers several rows of the table.
We found there are the following two sub-cases (also see Fig. 9).

  - **Sub-case 1:** The table begins with an anchor.
  - **Sub-case 2:** The table begins with texts.

  We found that in Sub-case 1, the <TR> tag before the anchor and the </TR> tag before the next anchor can divide the LSP from the other text portions. We found that in Sub-case 2, the <TR> tag after the previous anchor and the </TR> tag after the anchor can divide the LSP from the other text portions.

#### 4.2.3 Summary of HTML Tags for Dividing LSPs from the Other Text Portions

We found there are three kinds of HTML-tag set which can divide LSPs from the other text portions in original pages: the set including only the parent tag **(parent-tag set)**, the set including only the sibling tag **(sibling-tag set)**, and the set including the ancestor tag without the parent tag or both the parent tag and its sibling tag **(relative-tag set)**.

A parent-tag set consists of the parent tag which directly includes the anchor. Using the parent-tag set can divide a LSP from the other text portions when the LSP covers the whole of the paragraph, list item, table cell, or DIV object. For example, when a LSP covers the whole paragraph, the LSP can be divided by the <P> tag and </P> tag of the paragraph.

A sibling-tag set consists of the sibling tag which is at the same level as the <A> tag of the anchor in the document structure. Using a sibling-tag set can divide a LSP from the other text portions when the LSP is one part of the paragraph, list item, table cell, or DIV object. For example, when a LSP is one part of the paragraph which includes the anchor, the LSP is divided from the other text portions by the two sibling tags which are the <BR> tag before the LSP and <BR> tag after the LSP.

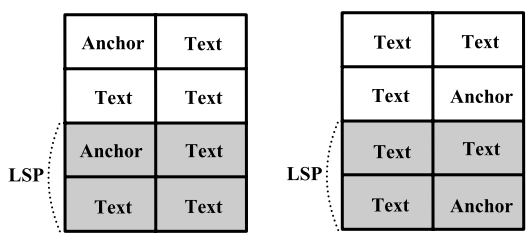



**Fig. 9** Two sub-cases in which the LSP covers several rows of a table.

A relative-tag set consists of either the ancestor tag without the parent tag or the both of the parent tag and its sibling tag in the document structure. Using a relative-tag set can divide a LSP from the other text portions when the LSP covers several cells (not all cells) of the current row, the current row, or several rows of the current table. For example, when a LSP covers several cells of the current row and the current row begins with an anchor, it is divided by the <TD> tag which is the parent of the anchor to the target page and the </TD> tag of the next cell which is its sibling tag. Furthermore, using a relative-tag set can divide a LSP from the other text portions when the LSP covers the definition term including the anchor and the definition description of the definition term.

Table 2 shows the numbers of LSPs which can be divided from the other text portions by using each type of tag set.

### 4.3 Survey of USP

This subsection explains the result of the survey about which kind of location the USP exists and what kind of HTML tag can divide the USP from the other text portions. Table 3 shows its result. The left column shows the type of upper-level object which is related to the anchor, the center column shows the number of pages which has each type of upper-level object, and the right column shows the HTML tags which can divide the USP from other text portions.

In our survey, we found 1097 original pages in which the page title is related to the anchor. There were 739 original pages in which headers (from H1 to H6) are related to the anchor. We also realized that if there are several headers at the same level (for example, there are several headers H3), the header nearest to the anchor is related to the anchor.

We found six original pages in which the table header of the current table is related to the anchor. We realized that authors of original pages rarely use table headers. They usually use the first row of the current table or the first row of the upper-level table instead of the table header. In 48 original pages, the fist row of the current table is related to the anchor; and in 82 original pages, the first row of the upper-level table is related to the anchor. We also found that the authors of original pages usually write some texts before the list as a list title. There were 64 original pages in which the text portion before the current list is related to the anchor. We realized that the numbers of words of these text portions are small. The biggest one among them is 19.

Unfortunately we found many STPs which cannot be extracted by HTML tags because their place cannot be iden-

**Table 2** Numbers of LSPs which can be divided from the other text portions by using each type of tag set.

| | Parent-tag set | Sibling-tag set | Relative-tag set |
|---|---|---|---|
| Paragraph | 216 | 102 | 0 |
| Ordered and un-ordered list | 329 | 25 | 0 |
| Definition list | 0 | 12 | 44 |
| Table | 165 | 63 | 113 |
| DIV | 21 | 18 | 0 |

**Table 3** Result of the survey of USP.

| Upper-level object | Total | HTML tags used for extracting STPs |
|---|---|---|
| Page title | 1097 | <Title> and </Title> |
| H1 | 326 | <H1> and </H1> |
| H2 | 209 | <H2> and </H2> |
| H3 | 153 | <H3> and </H3> |
| H4 | 18 | <H4> and </H4> |
| H5 | 26 | <H5> and </H5> |
| H6 | 7 | <H6> and </H6> |
| Table header | 6 | <TH> and </TH> |
| The first row of the current table | 48 | <TR> and </TR> |
| The first row of an upper-level table | 82 | <TR> and </TR> |
| The text portion before the current list | 64 | <BR> tag |
| Another row of the current table | 46 | cannot extract |
| Another row of the upper-level table | 167 | cannot extract |
| Another table | 278 | cannot extract |
| Another list | 36 | cannot extract |
| Another paragprah | 372 | cannot extract |

tified solely by HTML tags. We realized that, some of the authors of original pages write some texts related to the anchor in another row from the current row in the current table (not the row above the current row and not the first row in the current table). They also write some texts related to the anchor in a row of the upper-level table (not the row above the current table and not the first row in the upper-level table). We found 46 original pages with the former case and 167 original pages with the latter case. There were 278 original pages in which the text portion in another paragraph from the current paragraph (not the paragraph above or below the current paragraph) is related to the anchor. There were 36 original pages in which the text portion in another list from the current list is related to the anchor. There were 372 original pages in which the text portion in another table from the current table is related to the anchor. Currently, it is impossible to extract these text portions because this requires that the computer can semantically understand the content of the text.

## 5. Extraction of STP

In this section, we propose a method for extracting STPs based on the result of the survey of STP.

### 5.1 Extraction of LSP

Firstly, our method represents an original page by a DOM tree. It then identifies which location (paragraph, list item, definition list, table, or <DIV> object) the anchor to the target page belongs to. After that, the method extracts the LSP from the identified location. The detail of the method is as follows.

The method identifies which location the anchor belongs to according to the type of the parent tag as follows:

- <P>: the anchor is in a paragraph.
- <LI>: the anchor is in a list item.
- <DT> or <DD>: the anchor is in a definition list.
- <TD>: the anchor is in a cell of a table.
- <DIV>: the anchor is a DIV object.

Then the method extracts the LSP from each location as follows:

**a) If the anchor is in a paragraph, list item, definition object (<DD>) or DIV object:**
The method checks the number of <BR> tags in the parent object to the anchor.

- If there is no <BR> tag, it then extracts the whole texts of the object.
- If there is one or more <BR> tag(s), it then checks the number of anchors in the object. If there is only one anchor, it then extracts the whole text of the object. If there are several anchors, it then checks whether the object begins with an anchor or texts. If the object begins with an anchor, it extracts the text portion between the <BR> tag before the anchor and the <BR> tag before the next anchor. If the object begins with texts, it extracts the text portion between the <BR> tag after the previous anchor and the <BR> tag after the anchor.

**b) If the anchor is in a cell of a table:**
The method tries to expand to nearby cells by following the

left and right directions from the current cell. It repeats this expansion until it meets a cell which includes a different anchor. If it can expand to all cells of the current row which includes the current cell, it tries to expand to nearby rows by following the up and down directions. It repeats this expansion until it meets a row which includes a different anchor. There are the following four cases in the result of this expansion:

- **Case 1:** The method cannot expand to any other cells. The method extracts the LSP from the current cell by the same method as in a).
- **Case 2:** The method can expand to other cells but it cannot expand to all cells of the current row.
  The method then checks whether the current row begins with an anchor or texts. If it begins with an anchor, the method extracts the text portion between the <TD> tag before the anchor and the </TD> tag before the next anchor. If it begins with texts, the method extracts the text portion between the <TD> tag after the previous anchor and the </TD> tag after the anchor.
- **Case 3:** The method can expand to all cells of the current row.
  It extracts the whole texts in the current row.
- **Case 4:** The method can expand to other rows of the table.
  The method checks whether the table begins with an anchor or texts. If it begins with an anchor, the method extracts the text portion between the <TR> tag before the anchor and the </TR> tag before the next anchor. If it begins with texts, the method extracts the text portion between the <TR> tag after the previous anchor and the </TR> tag after the anchor.

**c) If the anchor is in <DT> object of a definition list:**
The method extracts the whole texts of the <DT> object and the whole texts of its <DD> object.

### 5.2 Extraction of USP

Our method extracts USPs as follows:

- It extracts the page title and all the upper headers from H1 to H6. If there are several headers at the same level, it extracts the nearest one to the anchor.
- It checks whether the anchor is in a table. If the anchor is in a table, it checks whether a table header exists. If a table header exists, the method extracts the table header. If a table header does not exist, the method checks whether or not the first row of the current table satisfies at least one of the following two conditions. (1) The number of its cells is smaller than the number of cells in the other rows. (2) There is no anchor in it while there are anchor(s) in all the other rows of the current table. If the first row of the current table satisfies at least one condition, the method extracts the first row of the current table. If the first row of the current table does not satisfy any condition, the method checks whether or not the first row of the upper-level table (if it exists) satisfies at least one of the above two conditions. If it satisfies at least one condition, the method extracts it. If it does not satisfy, the method continues to check the first row of the upper-level table of the previous upper-level table (if it exists). The method repeats this process until it finds out the first row which satisfies at least one condition or there is no more upper-level table.
- The method checks whether the anchor is in a list item. If it is in a list item, the method checks whether there is the following kinds of text portion before the list: (i) a text portion included in <P> tag, (ii) a text portion included in <DIV> tag and (iii) a text portion interleaved among two <BR> tags. If there is a text portion and its number of words is smaller than a threshold $\alpha$, the method extracts this text portion. We set $\alpha$ as 20 because in our survey of USPs, there is no list title which has the number of words which is greater than 19.

## 6. Evaluation of Extracted STPs

In the previous researches, they did not evaluate their methods from the viewpoint of extracting STPs. In our research, we invited three evaluators to participate in our experiments to give the real STPs. We evaluated the extracted text by using the correct answer of STP given by the evaluators. We also compared our method to other conventional methods in extracting STPs.

### 6.1 Dataset and Experimental Method

The dataset we prepared for our experiments contains 200 original pages. These pages are not included in the dataset explained in Sect. 4.1. These original pages were obtained by randomly selecting 10 official target pages and 10 personal target pages from Open Directory and collecting 20 original pages at most by Google. The average number of original pages per one target page is 10 exactly. The method to get the real STPs in each original page is as same as the one explained in Sect. 4.1.

We use precision, recall and the number of extracted words as evaluation parameter. The followings are the method for calculating precision and recall. We call the text portion extracted from the *i-th* original page by the extraction method $S_i$. $P_i$ is the STP of the *i-th* original page. Let |S| be the length of a text portion S (number of words in the text portion S). The $precision_i$ and $recall_i$ are calculated by the following equations:

$$precision_i = \frac{|P_i \cap S_i|}{|S_i|}$$

$$recall_i = \frac{|P_i \cap S_i|}{|P_i|}$$

The precision and the recall of the method when it extracts STPs from the dataset of 200 original pages are calculated by the following two equations:

**Table 4** Evaluation of our method for extracting STPs.

| | Precision | Recall | Average number of words of the extracted texts | Average number of words of word of the real STPs |
|---|---|---|---|---|
| LSPs | 97.01% | 93.94% | 20.36 | 21.07 |
| USPs | 89.43% | 74.35% | 8.54 | 9.35 |
| both of LSPs and USPs | 94.08% | 85.03% | 28.29 | 30.43 |

$$precision = \frac{1}{200} \times \sum_{i=1}^{200} precision_i$$

$$recall = \frac{1}{200} \times \sum_{i=1}^{200} recall_i$$

### 6.2 Evaluation of Our Method

We evaluated our method in extracting LSPs, USPs, both LSPs and USPs. Table 4 shows the experimental results. In this experiment, our method extracts LSPs in high precision (97.01%) and in high recall (93.94%). The number of words in the texts extracted as LSPs (20.36 words) is quite similar to the average number of words of the real LSPs (21.07 words). From this result, we can see that our method can identify the positions of LSPs in original pages accurately.

Our method extracts USPs in 89.43% precision and in 74.35% recall. The average number of words in the texts extracted as USPs (8.54 words) is almost same as the average number of the real USPs (9.35 words). These precision and recall are smaller than the precision and the recall in extracting LSPs. The reason why the precision of extracting USPs is smaller than that of extracting LSPs is explained as follows. Based on the result of the survey of USP, our method extracts the page title, the headers (H1 to H6), the first row of the current table, the first row of the upper-level table, and the text portion before the current list. However, in some original pages, these text portions are not related to the anchor. For example, some authors put the same name (in most cases, the name of the web site) to all pages. Some authors use headers or tables not for structuring the content of the document but for decorating the web page or creating the layout for the web page. This is why our method extracts noise keywords. The reason why the recall of extracting USPs is smaller than that of extracting LSPs is that there were some cases which our method cannot extract. As explained in Table 3, USPs in another row of the current table, another row of the upper-level table, another table, another list and another paragraph cannot be identified by HTML tags solely.

Our method extracts both LSPs and USPs in 94.08% precision and in 85.03% recall. The average number of words of extracted texts is 28.89. This is almost same as the average number of the real STPs (30.43 words). We do not know this precision and recall is high among other existing methods. The next subsection compares our method to the existing methods.

### 6.3 Comparison of Our Method to Other Methods in Extracting LSPs

We compare our method to the previous methods from the viewpoint of extracting LSPs.

#### 6.3.1 Previous Methods

The previous methods we compare to our method is the anchor-text method, fixed-window method, sentence-based method, paragraph-based method and list-based method. The anchor-text method extracts the text portion between the tags <A> and </A> of the anchor. The fixed-window method ([13] and [8]) extracts the anchor text and the predetermined number of words around the anchor. Both works [13] and [8] used 50 words before and after the anchor, because they thought that the appropriate size for the fixed-window is 50 words. We implemented the fixed-window method with three different options. Option 1 extracts 25 words before the anchor. Option 2 extracts 25 words after the anchor. Option 3 extracts 50 words including 25 words before the anchor and 25 words after the anchor.

The sentence-based method extracts one or more sentence(s) around the anchor. Delort's method [3] extracts the sentence containing the anchor. We implemented the sentence-based method with four different options. Option 1 extracts only the sentence containing the anchor. Option 2 extracts two sentences which are the sentence containing the anchor and the sentence before the anchor. Option 3 extracts two sentences which are the sentence containing the anchor and the sentence after the anchor. Option 4 extracts three sentences which are the sentence containing the anchor and the sentences before and after the above sentence.

The paragraph-based method extracts a paragraph which begins with the anchor followed by texts and there is no <BR> tag between the anchor and the texts ([5] and [10]). The list-based method extracts a list item which directly includes the anchor [9]. The paragraph-based method uses the <P> tag which is the parent object of the anchor. The list-based method uses the <UL>, <OL> or <DL> tag which is the parent object of the anchor. We think that these methods can be generalized as the **object-based method** which extracts texts of the parent object of the anchor. We also compared our method to the object-based method.

#### 6.3.2 Result of Comparison

We compared our method to these previous methods in extracting LSPs. Table 5 shows the experimental results. Note

**Table 5** Comparison of our method to the previous methods in extracting LSPs.

| Method | Precision | Recall | Average number of words of the extracted texts |
|---|---|---|---|
| Our method | 97.01% | 93.94 % | 20.36 |
| Anchor-text method | 100% | 44.31% | 3.43 |
| Fixed-window method (Option 1) | 24.22% | 56.31% | 26.96 |
| Fixed-window method (Option 2) | 43.85% | 79.51% | 27.61 |
| Fixed-window method (Option 3) | 29.52% | 91.52% | 51.38 |
| Sentence-based method (Option 1) | 100% | 56.13% | 6.73 |
| Sentence-based method (Option 2) | 58.19% | 61.11% | 14.18 |
| Sentence-based method (Option 3) | 78.17% | 82.95% | 16.29 |
| Sentence-based method (Option 4) | 60.1% | 89.7% | 25.54 |
| Paragraph-based method | 71.23% | 27.39% | 7.16 |
| List-based method | 87.24% | 35.08% | 7.92 |
| Object-based method | 70.95% | 87.53% | 367.45 |

∗Average number of words in real LSPs is 21.07

that the recall is calculated for extracting only LSPs not for extracting both LSPs and USPs.

The anchor-text method extracts LSPs in 100% precision because the anchor text is always related to the anchor. The recall of the anchor-text method is low (44.31%). This recall shows that 44.31% of real LSPs are anchor texts. The fixed-window method extracts LSPs in lower precision compared to the precision of the anchor-text method (24.22% (with Option 1), 43.85% (with Option 2) and 29.52% (with Option 3)). The recall of the fixed-window method is higher than the recall of the anchor-text method (56.31% (with Option 1), 79.51% (with Option 2), 91.52% (with Option 3)). We realize that the precision and the recall of the fixed-window method with Option 2 are higher than the fixed-window method with Option 1. It proves that people tend to write the explanation about the anchor after the anchor rather than before the anchor.

Because the sentence containing the anchor was always related to the anchor, the sentence-based method with Option 1 extracts LSPs in 100% precision. The recall of the sentence-based method is 56.13%. This shows that 56.13% of real LSPs are the sentences containing the anchors. We also realize that with Option 3, the sentence-based method has higher precision and higher recall than with Option 2. It proves again that people tend to write the explanation about the anchor after the anchor rather than before the anchor.

The paragraph-based method and the list-based method extract LSPs in lower precision (71.23% and 87.24%) than ours. It is because some of the paragraphs or list items are big and have several anchors. The recall of the paragraph-based method and the recall of the list-based method are low (27.39% and 35.08%). It is because in original pages, the anchor exists not only in a paragraph or in a list item but also in other kinds of places such as table and <DIV> object.

The object-based method uses more kinds of object than the paragraph-based and list-based method. Therefore its precision becomes lower (70.95%) and its recall becomes higher (87.53%) than them. Because some of the pages in the dataset are long and have no (or little) object, or have a long object with many anchors, the average number of words in the extracted texts becomes extremely high (367.45). Because some of the LSPs spreads between some objects, the recall of this method is lower than that of the fixed-window method with Option 3.

Our method solves these problems. For a big object including many anchors, our method extracts only one part of the object by using sibling tag. Therefore, our method achieves higher precision (97.01%) than that of the object-based method. When the anchor is in a table, our method not only extracts the cell which directly includes the anchor but also extracts text portions in nearby cells of the current row by checking the existence of anchors in nearby cells. Therefore, our method achieves higher recall (93.94%) than that of the object-based method. This recall is also the highest recall among all the previous methods.

### 6.4 Extraction of USPs

We compare our method to the previous methods from the viewpoint of extracting USPs.

#### 6.4.1 Previous Methods

The most basic method for extracting USPs is a method which extracts all upper-level objects of the anchor. Our method and Roy's method [6] extract the page title and headers (from H1 to H6). If there are several headers at the same level, they extract the nearest header to the anchor. The difference between two methods is as follows. Roy's method extracts the nearest decorated text portion to the anchor like strong, bold, italicized, or emphasized if this text portion is in the upper-level structure and there is no header between this text portion and the anchor. Our method extracts the first row of the current table, the first row of an upper-level table, and the text portion before the current list.

#### 6.4.2 Result of Comparison

We implemented Roy's method and the method which extracts all the upper-level objects. We compared our methods to these methods in extracting USPs. Table 6 shows the experimental results. Note that the recall is calculated for extracting only USPs not for extracting both LSPs and USPs.

**Table 6** Comparison of our method to the previous methods in extracting USPs.

| Method | Precision | Recall | Average number of words of the extracted texts |
|---|---|---|---|
| Our method | 89.43% | 74.35% | 8.54 |
| Extracting all upper-level objects | 13.01% | 100% | 1081.71 |
| Roy's method | 84.17% | 54.58% | 5.89 |

∗Average number of words in the real USPs is 9.35

The method which extracts all the upper-level objects extracts USPs in 100% recall. However, the precision is very low (13.01%) because the average number of words in extracted texts is very big (1081.71 words). The precision of our method (89.43%) is higher than that of Roy's method (84.17%). Roy's method extracts the nearest decorated text portion to the anchor. Some decorated text portions exist in tables. They sometimes go beyond cells from the current cell. Therefore some of them are not related to the anchor and the precision becomes worse than our method. Our method extracts the first row of the current table, the first row of an upper-level table, and the text portion before the current list. Therefore the recall becomes better than that of Roy's method.

### 6.5 Further Discussion

This section compared various kinds of method for extracting STPs. We should decide which method to use according to the upper-level application and the quantity of information (the number of original pages) which originally exists in the Web.

When the upper-level application is summarization, extracted STPs are shown to users as they are extracted although similar sentences are combined to one sentence. When the number of original pages is large, it is better to delete uncertain sentences (sentences which may not explain the target page) because it is troublesome for the user to read a large summary. In this case, we should select the anchor-text method or sentence-based method whose precision is 100%. When the target page has a few original pages, the user will appreciate to see related information as much as possible. In this case, we should select our method whose recall is the highest among existing methods.

When the upper-level application is categorization, the system does not show the extracted STPs as they are to the user. It makes a feature vector from words in the STPs and conducts a machine learning for making a model to judge whether or not another target page can be included in a category. Although it seems that we can make a good directory by using STPs extracted from the method with high precision like the anchor-text method and sentence-based method, Glover proved that we cannot achieve a good precision and recall for the categorization by using those methods [8]. This is because we do not have enough information in STPs extracted from the anchor-text method and sentence-based method. We should select a method which has a better balance between precision and recall. Thus we can expect that our method achieves the higher precision and recall for the categorization than the other methods.

As explained above, we can say that we should select a method for extracting STPs according to the type of process to realize the upper-level application and the quantity of information which exists in the Web. When the system shows the extracted text as it is to the user and the quantity of existing information is large, we should select a method for extracting STPs with high precision. Otherwise, we should select a method which has a good balance between precision and recall.

## 7. Evaluation of Extracted STPs for the Upper-Level Application

Although Sect. 6 proves that our method extracts STPs in high precision and recall, we are not sure that how much difference the upper-level application produces in actual users' usages. In this section, we applied our method to summarization as an upper-level application. We summarized the original pages to one target page. We compared the summaries created by our method to the summaries created by the most basic existing method and original pages as themselves in the user experiment where users should work on a specific task.

### 7.1 Experimental Method

We selected fixed-window method as the most basic existing method. We also compared our method and fixed-window method to original pages. In this experiment, the user should select one target page among five target pages from the user's objective. For creating real situations in which the user needs to select a web page from several choices, we designed five tasks. We also include many types of tasks to this experiment to know whether extracted STPs can be used for various objectives. The tasks are as follows:

**(1)** Find one page including evidence or statistic data about e-commerce among all pages about e-commerce.
**(2)** Find one page including explanation for beginners about eigenvalue decomposition among all pages about eigenvalue decomposition.
**(3)** Find one page including author's opinion to electronic appliances among all pages about electronic appliances.
**(4)** Find one page including story or article about travel among all pages about travel.
**(5)** Find one page explaining Java network programming among pages about Java programming.

Five original pages are used for each target page. The user cannot see the target page. Presentation of the summary differs by methods. In our method, page title, USPs other than page title, anchor text, LPSs other than anchor text is displayed independently (see Fig. 10). In fixed-window method, 50 words before and after the anchor is displayed (see Fig. 11). In original pages, the user should display pages one by one in different windows. The anchor to the target page is strongly visualized.

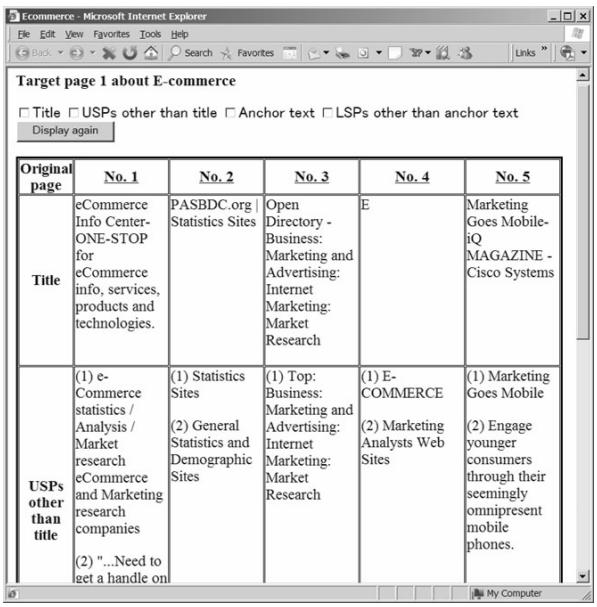
Experimental parameters are required time to finish the task and error ratio. Required time is calculated in each task. For calculating error ratio, the experimenter judged whether or not the target page the user selected is the page that the user should read. Error ratio is calculated by considering the judgments of all tasks.

15 users participated in this experiment. The users are divided into three groups equally. Although we selected 15 users who are in the same university and in the same age, there are individual differences among them in their English skills. For taking the counterbalance for this difference, we asked each group to work on the tasks as in Table 7.

### 7.2 Result

Error ratios of all tasks are 0% in three kinds of summaries. When users took enough time to judge whether or not they should read the target web page, they found the right page. On the contrary, required time differs in all tasks between our method, fixed-window method and original pages. The result is shown in Fig. 12. Apparently the user can judge whether to read or not to read more quickly from the summaries created by our method than from those created by fixed-window method. We conducted ANOVA (analysis of variance) and found a significant difference (F-value=394.72, p=0.00*, *p<0.01). The required time in every task is almost same and average time for all tasks is 183[sec] in our method, 926[sec] in fixed-window method, 1212[sec] in original pages. From this result, our method creates a summary which helps users' decisions in various objectives.

We found that the difference between our method and fixed-window method is large, but the difference between fixed-window method and original pages is not so large. The reason is that there were some cases the users should read the original pages when they read the summaries created by fixed-window method. When we invited three evaluators to judge which text portion is a real STP in the 125 original pages used in this experiment. The method for judgment is as same as the one explained in Sect. 4.1. We found 393 STPs from this judgment. We also asked three evaluators to judge each STP and each fixed-window text include conclusive expressions to find the answer target page in the given task. These judgments were also conducted by majority vote. Table 8 shows the numbers of LSPs, USPs,


Ecommerce - Microsoft Internet Explorer

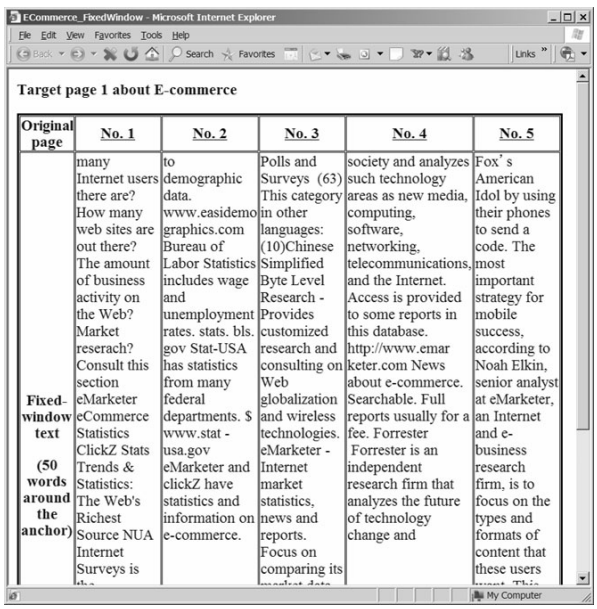
Target page 1 about E-commerce

☐ Title ☐ USPs other than title ☐ Anchor text ☐ LSPs other than anchor text

Display again

| Original page | No. 1 | No. 2 | No. 3 | No. 4 | No. 5 |
|---|---|---|---|---|---|
| Title | eCommerce Info Center-ONE-STOP for eCommerce info, services, products and technologies. | PASBDC.org \| Statistics Sites | Open Directory - Business: Marketing and Advertising: Internet Marketing: Market Research | E | Marketing Goes Mobile-iQ MAGAZINE - Cisco Systems |
| USPs other than title | (1) e-Commerce statistics / Analysis / Market research eCommerce and Marketing research companies<br><br>(2) "...Need to get a handle on | (1) Statistics Sites<br><br>(2) General Statistics and Demographic Sites | (1) Top: Business: Marketing and Advertising: Internet Marketing: Market Research | (1) E-COMMERCE<br><br>(2) Marketing Analysts Web Sites | (1) Marketing Goes Mobile<br><br>(2) Engage younger consumers through their seemingly omnipresent mobile phones. |

**Fig. 10** Summaries created by our method.


ECommerce_FixedWindow - Microsoft Internet Explorer

Target page 1 about E-commerce

| Original page | No. 1 | No. 2 | No. 3 | No. 4 | No. 5 |
|---|---|---|---|---|---|
| Fixed-window text (50 words around the anchor) | many Internet users there are? How many web sites are out there? The amount of business activity on the Web? Market reserach? Consult this section eMarketer eCommerce Statistics ClickZ Stats Trends & Statistics: The Web's Richest Source NUA Internet Surveys is | to demographic data. www.easidemographics.com Bureau of Labor Statistics includes wage and unemployment rates. stats. bls. gov Stat-USA has statistics from many federal departments. $ www.stat - usa.gov eMarketer and clickZ have statistics and information on e-commerce. | Polls and Surveys (63) This category in other languages: (10)Chinese Simplified Byte Level Research - Provides customized research and consulting on Web globalization and wireless technologies. eMarketer - Internet market statistics, news and reports. Focus on comparing its | society and analyzes such technology areas as new media, computing, software, networking, telecommunications, and the Internet. Access is provided to some reports in this database. http://www.emarketer.com News about e-commerce. Searchable. Full reports usually for a fee. Forrester Forrester is an independent research firm that analyzes the future of technology change and | Fox's American Idol by using their phones to send a code. The most important strategy for mobile success, according to Noah Elkin, senior analyst at eMarketer, an Internet and e-business research firm, is to focus on the types and formats of content that these users |

**Fig. 11** Summaries created by fixed-window method.

**Table 7** Counterbalance for individual differences.

| | Group A | Group B | Group C |
|---|---|---|---|
| Task 1 | Our method | Fixed-window method | Original pages |
| Task 2 | Fixed-window method | Original pages | Our method |
| Task 3 | Original pages | Our method | Fixed-window method |
| Task 4 | Our method | Fixed-window method | Original pages |
| Task 5 | Fixed-window method | Original pages | Our method |

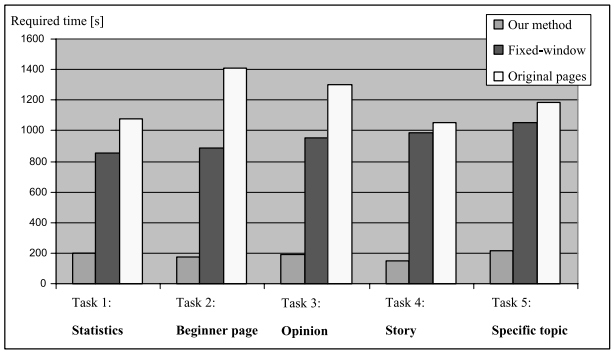



**Fig. 12** Required time for each task when using three kinds of summaries.

**Table 8** The number of text portions including conclusive expressions.

| | |
|---|---|
| The number of extracted LSPs including conclusive expressions | 19 |
| The number of extracted USPs including conclusive expressions | 38 |
| The number of fixed-window texts including conclusive expressions | 13 |

fixed-window texts which include conclusive expressions. We found that the number of extracted text portions which include conclusive expressions is smaller when using the fixed-window method than when using our method. This leads to the result that the user should read original pages when using the fixed-window method.

From this experiment, we found that there are more conclusive expressions in USPs rather than LSPs and fixed-window texts for judging whether or not the user should read the target page in various kinds of tasks. This decreased the required time for the user to make a decision when using our method. We applied the extracted text portions by our method only to the summarization. However because we found a significant difference in time to complete tasks, we think that our extraction method can be useful for other upper-level applications.

## 8. Conclusion and Future Work

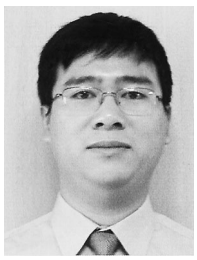
This paper concentrates on extracting a semantic text portion (STP) from an original page. STP is a text part which is related to the anchor to the target page. Firstly, we found two types of STP: local semantic portion (LSP) and upper-level semantic portion (USP). We conducted a survey for each type of STP by using 1108 real original pages to find HTML tags which can semantically divide STPs from the other text portions in original pages. We invited three evaluators to participate in our survey to judge which text portions in an original page are STPs. We then developed a method for extracting STPs based on the result of the survey. Our method represents an original page by a DOM tree to analyze its document structure. It then extracts STPs by using specific set of HTML tags which are found in the survey. We then conducted experiments to evaluate our method and compare it to the previous methods in extracting STPs. We evaluated the texts extracted by each method by comparing them to the real STPs given by three evaluators. The experimental results showed that our method achieves high precision and the highest recall compared to the previous methods. Finally we applied our method to summarization. When we conducted a user experiment where the user should work on several real tasks, the summaries created by our method helps users to make a decision.

The shortcoming of our survey and our extraction method is that they just consider the relevance to the anchor but do not consider the type of relevance. We found in the survey that STPs are either facts or people's opinions (evaluation and categorization). In some applications, we should select the type of STPs. In the summarization, the user may want to see only the people's evaluation. In the categorization, the user may want to see a categorization created by a user group from their viewpoints. We will study STPs by considering whether they are facts, people's evaluation or people's categorization as a future work.

### References

[1] B.D. Davison, "Topical locality in the Web," Proc. 23rd Annual International Conference on Research and Development in Information Retrieval (SIGIR 2000), pp.272–279, 2001.
[2] M. Henzinger, "Link analysis in Web information retrieval," IEEE Data Engineering Bulletin, vol.23, no.3, pp.3–8, 2000.
[3] J. Delort, B.B. Meunier, and M. Rifqi, "Enhanced Web document summarization using hyperlinks," Proc. 14th ACM Conference on Hypertext and Hypermedia (HT'03), pp.208–215, 2003.
[4] E. Amitay, "Using common hypertext links to identify the best phrasal description of target web documents," Proc. SIGIR'98 Post-Conference Workshop on Hypertext Information Retrieval for the Web, pp.271–276, 1998.
[5] E. Amitay and C. Paris, "Automatically summarizing Web sites: Is there a way around it?," Proc. ACM 9th International Conference on Information and Knowledge Management (CIKM 2000), pp.173–179, 2000.
[6] S. Roy, S. Joshi, and R. Krishnapuram, "Automatic categorization of websites based on source type," Proc. 15th ACM Conference on Hypertext & Hypermedia, pp.38–39, 2004.
[7] M. Otsubo, B.Q. Hung, Y. Hijikata, and S. Nishida, "A basic study on Web page classification method by anchor-related text," Proc. SICE Annual Conference 2005, The International Conference on Instrumentation, Control and Information Technology, pp.3622–3625, 2005.
[8] E.J. Glover, K. Tsioutsiouliklis, S. Lawrence, D.M. Pennock, and G.W. Flake, "Using Web structure for classifying and describing web pages," Proc. 11st International World Wide Web Conference, pp.562–569, 2002.
[9] G. Attardi, S. Di Marco, and D. Salvi, "Categorisation by context," J. Universal Computer Science, vol.4, no.9, pp.719–736, 1998.
[10] Furnkranz, "Exploiting structural information for text classification on the WWW," Proc. 3rd Symposium on Intelligent Data Analysis (IDA-99), pp.487–498, 1999.
[11] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," Proc. 11st Annual Conference on Computational Learning Theory, pp.92–100, 1998.
[12] DOM, http://www.w3.org/DOM/
[13] S. Chakrabarti, B. Dom, D. Gibson, J. Keinberg, P. Raghavan, and S. Rajagopalan, "Automatic resource list compilation by analyzing hyperlink structure and associated text," Proc. 7th International World

Wide Web Conference, pp.65–74, 1998.
[14] J.M. Kleinberg, "Authortitative sources in a hyperlinked environment," J. ACM, vol.46, no.5, pp.604–632, 1999.
[15] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," Proc. 7th International Conference on World Wide Web, pp.107–117, 1998.
[16] Open Directory, http://dmoz.org/
[17] Google, http://www.google.com/



**Bui Quang Hung** was born in Thai Binh, Vietnam in 1979. He received his B.Sc. from Vietnam National University in 2001 and M.E. from Osaka University in 2005. Currently, he is a Ph.D. candidate in Division of Systems Science and Applied Informatics, Graduate School of Engineering Science, Osaka University. His research interests are on text mining, information extraction and Web technology.

**Masanori Otsubo** was born in Fukuoka, Japan. He received his B.E. from Osaka University in 2005. Currently, he is a master course student in Division of Systems Science and Applied Informatics, Graduate School of Engineering Science, Osaka University.

**Yoshinori Hijikata** was born in Kobe, Japan. He received the B.E. and M.E. degrees from Osaka University in 1996 and 1998, respectively. In 1998, he joined IBM Research, Tokyo Research Laboratory. After working on Web technologies there, he received Ph.D. degree from Osaka University in 2002. Currently, he is a research associate in Osaka University. His research interests are on Web intelligence, personalization and text mining. He received the best paper awards from IPSJ Interaction'05 and ACM IUI'06. He is a member of the IPSJ, JSAI, HIS, DBSJ and IEEE.

**Shogo Nishida** was born in Hyogo Prefecture in 1952. He received the B.S. M.S. and Ph.D. degrees in Electrical Engineering from the University of Tokyo, in 1974, 1976 and 1984, respectively. From 1976 to 1995, he worked for Mitsubishi Electric Corporation, Central Research Laboratory. From 1984 to 1985, he visited MIT Media Laboratory, Boston, Massachusetts, as a visiting researcher. In 1995, He moved to Osaka University as a Professor of Graduate School of Engineering Science. His research interests include CSCW, Media Technology, Human Interfaces and Human Communication. He is a Fellow of IEEE (1998), and is currently the President of Human Interface Society in Japan (2004–2005). He received the achievement award in 2004 , the best paper awards in 1986 and 1993, the best book award in 1992 and the progress award in 1995 from IEE in Japan. He also received the best paper awards in 2001 and 2005 from Human Interface Society in Japan.