Abstract

In this paper, we propose a novel method for generating personalized page links. The page links which are generated by our proposed method are useful if users look for web pages related to the current web pages. To generate these page links, we proposed the following two steps, such as 1) divide current web pages into significant minimal blocks, and 2) find web pages related to these blocks. We confirmed in our experimental evaluation that users can find web pages related to the users’ interests.

1. Introduction

Recently, when users browse web pages, these users generally use keyword-based web search engines, such as Google and Yahoo! Then, the users browse the results of web search engines by clicking. If the results are different from the users’ interests, the users add or remove keywords, and gain the results of web search engines again. In this way, the behavior of clicking page links is one of the important feature to replace the function of keywords. This is because, we believe that users are not always select appropriate keywords, and selecting page links is easier for users than selecting appropriate keywords.

On the other hand, web page publishers create web pages which generally consist of page links. One of the aims of page links is for navigating users to appropriate web pages. However, these page links are not always useful for users to reach appropriate web pages. The reason of this problem is that, every users do not always need same page links whereas the publishers are not always create adequate page links. Therefore, personalized page links should be useful to navigate users to appropriate web pages.

In this paper, we propose an automatic page link generation method to retrieve the users’ desired web pages by only selecting generated page links and original page links. First, as same as general keyword-based search engines, users input some keywords to our proposed system. Then, our system returns one web page as a retrieval result. This result is close to Google’s “I’m feeling lucky” function. However, our system adds several automatic generated page links to the web page. The link targets are relevant to a part of the result web page, and the places where the generated page links placed on the web page are identified based on a similarity between the link target and a part of the web page. When the users browse this result web page, the users can browse the detail information of the result web page, or the users can browse the other kinds of information by selecting generated information. Of course, the original page links are also useful to search information related to the users’ interest. We believe that original page links on web pages are not enough to navigate the users to the information which the users interests. Therefore, to add page links automatically, depends on the users’ keywords, the users can browse the information easily.

Our proposed method has two features. One feature is a selection method of link targets. In this study, we select link target web page as similar web pages to the retrieved web page. In general, many retrieval systems are able to search web pages similar to the retrieved web page using their own algorithms. However, these retrieved web pages are also similar with each other. Therefore, when we use these algorithms, the link targets the system select are selected as many similar web pages. We assume that the users’ page link selection should be considered as the users’ queries. From this reason, to give many options of link targets to users, we believe that the selected web pages as link targets should different with each other. In this paper, we catego-
rize the web pages which are similar to the retrieved web page into several groups, and select main web page from each group.

Another feature is the place to conclude page links in the result web pages. When we add page links to the retrieved web page, we should select the place of the page links. In general, many web pages have multiple topics in one page. Therefore, when the topics of the link target and the topics of a part of retrieved web pages are different, users should confuse why these generated page links are placed. Then, we discuss an appropriate position of generated page links.

To confirm the capability of our proposed methods, we did an experimental evaluation. In our experiments, we compared the number of page clicks when the users use our proposed system with that when the users use Google. From our experimental results, our system can reduce the number of page clicks to be half.

We describe our proposed method as follows. In section 2., we describe about keyword-based search engine, and discuss why our proposed method, generation of page links, are better than ranked lists. Next, in section 3., we describe the method of generating page links. Then, in section 4., we show an experimental results and discussion. In section 5., we show the related works and the difference between our proposed method and their works. Finally, in section 6., we conclude this paper.

2. Keyword-based Search Engine

In this section, we describe about related works about this research.

In keyword-based retrieval system, when users input keywords to a retrieval system, the system returns web pages which are related to users’ keywords. The retrieval results are expressed using ranked web page lists. Therefore, when the number of retrieval results become large, users should browse many retrieval results to find appropriate information. However, browsing numbers of web pages is hard for users. On the other hand, if the retrieval system does not return any result, the users should change the keywords, and retrieve again. In this way, keyword-based retrieval systems have problems about users’ input.

Therefore, to avoid the problem of keyword-based retrieval system, we proposed a click-based retrieval system. In our proposed system, users input keyword and clicks. The click means that, the link anchor which users click affect the results of click-based retrieval system. Then, if users input same keywords to our retrieval system, but the users have different interests, the results of the retrieval system change based on users’ selection of link anchor.

3. Automatic Page Link Generation

In this section, we describe the details of our automatic page link generation method. When we generate page links, we should consider two issues, such as the position of page links and the direction of web pages. In our proposed method, the system process web pages using the following six steps.

1. Using keywords which users input, the system get retrieval results from keyword-based web search engines.
2. The system access all web pages in the retrieval results.
3. The system divides these web pages into the significant sections to identify the start point of generated page links.
4. The system calculates the similarity values between the significant sections and many web pages on the Internet to identify the end point of page links.
5. The system identify the page links based on the similarity values calculated before, and generate page links.
6. The system returns the retrieval results from the keyword-based web search engines. Each result has web page with generated page links.
In section 3.2, we describe the step 3., the method to divide web pages into significant sections. Next, in section 3.3, we describe the step 4. the method to calculate the similarity values between the significant sections and the web pages.

In step 4., if the system has unlimited bandwidth network and unlimited CPU resources, the system can calculates all web pages on the Internet. However, general computers have limitations about network bandwidth and CPU resources. In this paper, we assume that the web pages are on one web site, such as a company’s site or a college’s site, and the number of web pages should be limited, small number.

### 3.1 Term Weighting Scheme

In this section, we define a term weighting scheme to measure the similarity values between significant section and web page. In this paper, we use TF-ICF [10] as the term weighting scheme instead of TF-IDF. TF-ICF is the weighting scheme for categorized web pages. In TF-IDF method [1, 13, 14], the weights of terms are calculated using term frequency (TF) and inverse document frequency (IDF). In this term weighting scheme, TF means the comprehensiveness of a term, and IDF means the rareness of a term. Therefore, the term weight of TF-IDF is large if the term is comprehensive and also rare. However, web pages have web page structures, and each web page have an relationship between the web page and the other web pages. When we categorize web pages into groups using existing page links, the accuracy of term weights using TF-ICF are better than that using TF-IDF.

Here, we define the weight \( w_{ij} \) of the term \( k_j \) in function 1. When we have term \( k_j \) which is written in web page \( p_i \), the weight of weight is \( w_{ij} \), term frequency of \( k_j \) is \( T_{ij} \), inverted category frequency is \( C_j \), term frequency of \( k_j \) in web page \( p_i \) is \( f_{ij} \), the number of categories is \( O \), and the number of category which consists of the term \( k_j \) is \( c_j \), the the weight \( w_{ij} \) is defined as the following function.

\[
\begin{align*}
  w_{ij} &= T_{ij} \cdot C_j \\
  &= f_{ij} \cdot \left( \log \frac{O}{c_j} + 1 \right)
\end{align*}
\]  

(1)

In section 3.1.1, we show how to generate web page categories.

#### 3.1.1 Generation of Web Page Categories

In this section, we describe how to categorize web pages to calculate TF-ICF.
1. When any two web pages are on the same directory, these two pages are categorized on the same group.

2. When any two web pages are on the different directory, but one directory is a parent directory of another directory, these two pages are categorized on the same group.

3. However, if any two pages fulfill the previous two conditions, if the directory has main page, such as index.html and default.htm, the web pages on the directory and that on the other directory are separated into the other groups.

Here, the word “main page” means the index page of the directory, such as index.html, top.html, main.html, and home.html.

We describe the method of generating categories of web pages. Here, we define the number of category $c_k$ which consist of term $k_u$. All categories are not overlapped. $O$ means the number of all categories.

1. First, we identify the name of category of web page. To get the name of category, we get URI of the web page. And then, we remove the file name from the URI.

2. If this category does not have main page, we check the parent directory whether main page exists or not. If main page exists, we generate a category which consist of web pages in these directories.

3. If this category has main page, we generate a category which consists of the directory.

3.2 Start Point of Page Links

In this section, we describe how to select position of page links as start point. Before describe our proposed method, we define some notions used in this paper.

First, we have a web pages $D$ which are retrieved by the keyword-based web search engine. We identify the start point of page links using the following instruction.

1. The system extracts terms, and the position of terms from $D$.

2. The system also extracts HTML tags and the position of tags from web page $D$.

3. To divide web page $D$ into meaningful sections, the system first divide $D$ using HTML tags. We used $< p >$ tags to divide $D$ into meaningful sections $d_1(D), d_2(D), \ldots, d_O(D)$.

4. To reduce the sections, the system combines these sections if two of any sections have similar contents.

From this instruction, we focus on how to divide web page into meaningful sections. In the following section, we describe the definition of the meaningful section in this paper, and method of division.
3.2.1 Concept of Meaningful Section

In this section, we describe how to separate web page into meaningful sections.

When we generate page links to the related link targets, we should consider the place of the generated page links in order to find appropriate page links easily. However, web pages are generally written using HTML. Generally, HTML tags are not always indicate semantic structures of the web pages, because HTML tags are often used to express graphical structures. For example, table tags are often used to express the layouts of web pages. Therefore, dividing web pages using only HTML tags are not useful.

In this paper, we used the dense of the terms and HTML tags to divide meaningful sections. We assume that if there are two paragraphs on a web page, and these paragraphs has many same terms, the topics of these paragraphs are the same, and these paragraphs should not separate. However, if two paragraphs have different terms, these paragraphs should separate.

In section 3.2.2, we describe a method of dividing web pages in detail.

3.2.2 Dividing Method of Web Pages

1. The system divides the web page \( p_i \) \((i = 1, 2, \ldots, N)\) into terms, and extracts feature terms, frequently appeared noun.

2. The system extracts \( M \) kinds of terms in the web page \( p_i \), and calculates the average position their standard deviation of each term.

3. When the average interval of the terms is greater than the parameter \( E \), then the system extract the terms as separating terms. If a term is appeared once in the web page, we do not extract this term.

4. The system scans the web page again from top. When separating terms appear, the system find HTML tags which are close to the separating terms. The system identify the HTML tag as separating tags, and divide the web page using the separating tags.

We describe in step 2 using the example in Figure 4. In this example, we focused on the term pen. The term pen is appeared three times, and appeared in 4-th, 8-th, and 11-th position. Then, the average position of term pen is \((4 + 8 + 11)/2 = 7.7\). The standard deviation of this term is \(\sqrt{(7.7 - 4)^2 + (7.7 - 8)^2 + (7.7 - 11)^2}/14 \approx 4\). When the system set a parameter \( E \) is set to 3, this term pen is a separating term, because the standard deviation of pen is greater than the parameter \( E \). Next, we select two pen which interval is greater than \( E \). We can recognize that second and third pen has intervals greater than \( E \), then the first \(<br/>\) tag is selected as separating tag. Then we separate first sentence from second and third sentence.

<table>
<thead>
<tr>
<th>F</th>
<th>4. Ex</th>
<th>W b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag name</td>
<td>Meaning of Tag</td>
<td></td>
</tr>
<tr>
<td>(&lt;HR&gt;)</td>
<td>hairline</td>
<td></td>
</tr>
<tr>
<td>(&lt;Hn&gt;)</td>
<td>title</td>
<td></td>
</tr>
<tr>
<td>(&lt;p&gt;)</td>
<td>paragraph</td>
<td></td>
</tr>
<tr>
<td>(&lt;span&gt;)</td>
<td>block</td>
<td></td>
</tr>
<tr>
<td>(&lt;div&gt;)</td>
<td>block</td>
<td></td>
</tr>
<tr>
<td>(&lt;br&gt;)</td>
<td>line break</td>
<td></td>
</tr>
<tr>
<td>(&lt;table&gt;)</td>
<td>table</td>
<td></td>
</tr>
</tbody>
</table>

3.3 End Point of Page Links

In this section, we describe how to identify the end point of generating page links. We identify the end point of page links using the following methods.

1. Using keyword-based web search engines, the system inputs a section \( d_i \) of web pages \( D \) described in section 3.2, and retrieve web pages \( p_1, p_2, \ldots, p_n \) which have similar to \( d_i \).

2. The system generates clusters \( G_1, G_2, \cdots, G_L \) from web pages \( p_1, p_2, \cdots, p_n \). We define \( L \) as the number of clusters.
3. To identify the end point of generating page links, the system selects the most valuable web page from each cluster $G_k$. We define $P_k$ as the most valuable web page of the cluster $G_k$.

### 3.3.1 Similarities of Web Pages

In this section, we use vector space model [1] to calculate the similarities between two web pages. First, we describe a feature vector $\vec{P}_i$ of web page $p_i$ as equation (2).

$$\vec{P}_i = [w_{i1}, w_{i2}, \cdots, w_{iM}]$$

(2)

Then, we define the similarity values $e(p_x, p_y)$ ($x, y = 1, 2, \cdots, N$) is defined using similarity function $S$ as equation (3).

$$e(p_x, p_y) = S(\vec{P}_x, \vec{P}_y) = \frac{\sum_{j=1}^{M} (w_{xj} \cdot w_{yj})}{\sqrt{\sum_{j=1}^{M} w_{xj}^2} \cdot \sqrt{\sum_{j=1}^{M} w_{yj}^2}}$$

(3)

### 3.3.2 Selection of Link Targets

As we already described, we gain several groups of web pages from keyword-based web search engine. Each group has similar web pages. The users do not want to need similar web pages, then the system should not show these web pages. To solve this issue, we used the following method to select link targets from each group.

First, we count the term frequency of keyword which user inputs in each web page. We also count the similarities between the retrieved web page and each web page. Then, we add the term frequency and the similarities. Finally, we select highest values as link targets. The link targets are selected in each group. The number of link targets and that of the group is the same.

In the following sections, we denote these selected link targets as $E_j$ ($j = 1, 2, \cdots, D_k$).

### 3.4 Generation of Page Links

From section 3.2 and 3.3, we decide the start and end point of the generated page links. In this section, we describe how to generate page links.

First, we calculate similarity values between selected start point $p_i$ and end point $E_j$ using the following function.

$$Sim(p_i, E_j) = \frac{\vec{P}_i \cdot \vec{E}_j}{||\vec{P}_i|| \cdot ||\vec{E}_j||}$$

(4)

where $\vec{P}_i$ is the feature vector of $p_i$, and $\vec{E}_j$ is the feature vector of $E_j$.

In each $j$ of $E_j$, we select the most relevant $p_k$, and generate page link $L(p_k, E_j)$ as follows.

$$p_k = \{p_k | \max Sim(p_i, E_j)\}$$

(5)

Using the above method, the system automatically generate page links. In section 4, we confirm our method works well or not.

### 4 Experimental Evaluation

To confirm the capability of our proposed method, we did an experimental evaluation. In this evaluation, we evaluate how our proposed method generates appropriate page links, and how the number of clicks of users reduce. We cannot find similar concepts of study as related works. Therefore, we compared our proposed system with keyword-based web search engine Google[6].

#### 4.1 Experimental Setup

Before experiment, we proposed a set of web pages in our university. We prepared 2,500 web pages as page link targets. We assume that the users are interested in “barrier-free”, and the authors defined the following three pages as the answer sets.

- The top of page of Barrier-free Lab.
- An abstract of thesis related to barrier-free.
- An introduction of the research in Barrier-free Lab.

We also prepare the other four sets of keywords and answer sets, such as “welfare”, “the elderly”, ”sign language”, and “care”. These answer sets are generated by the graduate school students. The number of student is five.

We compare the number of clicks to pick up one of the web pages in the answer set.


<table>
<thead>
<tr>
<th>Input keyword</th>
<th>Generated Keyword</th>
<th>Original</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bareer-free</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>welfare</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>the elderly</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>sign language</td>
<td>-</td>
<td>33.33</td>
</tr>
<tr>
<td>care</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>5.8</strong></td>
<td><strong>13.5</strong></td>
</tr>
</tbody>
</table>

We describe how to count the number of access frequencies in detail. When users use our proposed method, the users first input the keywords, such as “barrier-free”, to our proposed system. Then, our system outputs one web page with generated page links. When the users click one of generated page links, our system outputs the other web page with generated page links again. Finally, the users can pick one of the web pages in the answer set. We count the number of frequency of access in the process for three users, for three times, respectively.

Next, when the users use keyword-based search engine, users also first input the keyword to the search engine. Then, the search engine returns 10 web pages. When there is no appropriate web page in this research, users click “next” button to browse the other 10 pages. We count the number of access in this process.

Finally, when users use original page links, users first input keywords to the keyword-based system. Then, the search engine returns one web page. The users only access page links on the web page to gain desired information. We count the number of access in this process.

4.2 Experimental Results and Discussion

We describe the result of experimental evaluation in Table 2. From this result, we show that the average clicks of our system is smaller than the other methods, such as the keyword-based search engine and no page link generation. Therefore, we confirm that using our proposed method, the average number of access frequency can reduce from 13.5 times to 5.8 times.

The reason is because, in this experiment, the keyword the user inputs are less than two terms. However, the users’ interests are more complex than the keywords. Therefore, users cannot fully describe the users’ interests as keywords. In this situation, this keyword-based web search engine is hard to search appropriate web pages. On the other hand, using our proposed method, the users can deal with both keywords and clicks. Then the system can retrieve and present appropriate web pages from the description which page links are clicked.

5 Related Works

In this section, we describe about automatic page link generation, and difference between these works and our proposed method. We also describe the other studies about dividing web pages.

5.1 Automatic Generation of Page Links and Web Pages

Kiyomitsu et al. [9, 8] proposes the method of dynamically changing the contents of web pages depends on the situation or environment of users. In this method, the system automatically generate web pages depends on the process speed of the users’ PC and the access speed. This authors also describe automatic changing method of web pages using the users’ access patterns. The retrieval results of these studies are similar to that of our method. However, these system generates page links based on the page constructors’ aim. In contrast, our proposed method generates page links based on the users’ interests.

Fukumura et al. [5] proposes a component, which is a group of contents and relation between web pages which users browse, for personalizing web search engines. In short, they propose a browsing history of users for customizing web search results. Our proposed method is similar to this method in the point view of personalizing web search engine. However, this proposed system do not place appropriate page links, only list page links.

5.2 Clustering of Retrieval Results

Clustering of retrieval results are main issue in the web search engine research field. Vivisimo [16] and GATA [7] are popular as the search engine which can clustering retrieval results. These systems returns retrieval result as similar web pages to the users’ keywords. However, in our proposed method, we generate different types of web pages in different clusters. These systems also do not create web
5.3 Dividing Methods of Web Pages

Cai et al. [2] proposes VIPS algorithm to divide web pages. In his study, web pages are divided using structure of HTML tags, and visual information such as background color and size of character. However, they use only HTML tag information, they do not consider terms. In this paper, we not only deal with HTML tag information but also deal with term frequency and density information.

5.4 Appearance Density of Terms

Sano et al. [15] proposed a method for scoring web pages using hunning window function. Our proposed method is similar to this method because these two methods are focused on extracting appearance density of terms. Sano use this method for scoring web page, our proposed method use this for dividing web pages.

6 Conclusion

In this paper, we proposed a novel method for generating page links automatically. Using our proposed system, users can browse web pages related to the users’ interests easily. If users cannot express the users’ interests as keywords, users are able to access relevant web pages by only clicking page links.

In this paper, we proposed two methods as follows:

- We propose a method for dividing web pages into significant parts. When we divide web pages, we used term density instead of term frequency. As a result, our system can place the generated page links into appropriate position.
- We propose a method for selecting appropriate web pages as link targets. We select the web pages which are similar to the web page but different with each other. As a result, the users do not need any similar web pages as link targets.
- We used X-means algorithm [11] for clustering web pages instead of K-means algorithm [12] when users select link targets. However, when we use this clustering algorithm, we should set some parameters to define the number of link targets. Therefore, we should develop a clustering algorithm to set appropriate parameters automatically.

- In our proposed method, we calculate similarities for all combination of web pages. Therefore, the calculation cost of our proposed method should be large. To reduce this calculation cost, we should use top-k ranking algorithm such as [2, 4, 3].

References


