Far East Journal of Experimental and Theoretical Artificial Intelligence
Volume … Issue …, 2008, Pages …
This paper is available online at http://www.pphmj.com

SUGGESTING SUBJECT HEADINGS USING WEB INFORMATION SOURCES

HIROSHI UEDA, HARUMI MURAKAMI[*] and SHOJI TATSUMI

Graduate School of Engineering
Osaka City University
3-3-138, Sugimoto, Sumiyoshi
Osaka 558-8585, Japan
e-mail: d06tb001@ex.media.osaka-cu.ac.jp

[*]Graduate School for Creative Cities
Osaka City University
3-3-138, Sugimoto, Sumiyoshi
Osaka 558-8585, Japan
e-mail: harumi@media.osaka-cu.ac.jp

Abstract

We proposed a method that suggests subject headings based on user queries when a pattern-matching algorithm fails to locate subject searches for Online Public Access Catalogs (OPAC). We combined information obtained from Wikipedia, Amazon, and Google for query expansion. Our method has two main advantages: (1) availability for any library without customizing OPACs, and (2) ability to suggest subject headings when a query string is not included in OPAC's bibliographic information. Three experimental results using computer terms revealed the following: (1) Suggested subject headings were related to the input term; (2) Suggested subject headings were better when we used a mixture of Wikipedia, Amazon, and Google than when just using one of them; (3) Our method can suggest subject headings when OPAC mining

Keywords and phrases: OPAC, subject search, subject headings, query expansion, Wikipedia, Amazon, Google.

Communicated by Shun-Feng Su

Received July 26, 2008; Revised September 11, 2008

cannot. We conclude that our method can serve as an alternative when pattern-matching algorithms fail.

1. Introduction

Subject search is one essential search function that enables users to search collections by subject information in library Online Public Access Catalogs (OPAC). Examples of subject information include author names, subject headings, and classification numbers. Among these, subject headings are controlled vocabularies and assigned to each collection (e.g., books, periodicals). When librarians register a collection with OPAC, they analyze its contents, select appropriate subject headings, and assign them to OPAC. Subject search is useful when titles do not accurately express the contents of the collection. Consider a book entitled “How to search for information in the internet age” to which subject heading “information retrieval” is assigned. When a user inputs “information retrieval” as a query, this book will be retrieved although the title does not contain the word “retrieval.” In addition, when a user selects a subject heading and performs a subject search, no book unrelated to information retrieval is included in the search results. Thus, the two great merits of subject search are that (1) a user can search through collections whose titles do not include the user’s query and (2) a subject search may well give precise results.

Most systems use pattern-matching algorithms to display subject headings based on user queries. However, when a pattern-matching algorithm fails, no subject heading is displayed. In this paper, we investigate a method that suggests subject headings when pattern-matching algorithms fail.

The following empirical procedure is carried out by librarians searching for subject headings. First, they perform a keyword search and get some results. Second, they make selections from those results. Third, they select subject headings contained in these results and perform a subject search. Automating this process may suggest subject headings. Red Light Green[1] [1] by Online Computer Library Center, one example of

[1] RedLightGreen service ended on December 2006.

such a system, displays a list of subject headings that are included in the search results of the keyword search. A secret to Red Light Green's success is its "good ranking of search results." It has a huge catalog of over 120 million books in worldwide libraries from which it derives its rankings. When a user inputs a query, it displays subject headings that are included in high-ranked search results. We call this method "OPAC mining" in this paper.

However, most OPACs are designed for only one library, and their order of search results is alphabetic. In addition, most libraries lack the budget to modify OPAC software packages developed by vendors. To implement such OPAC mining as RedLightGreen for suggesting subject headings, libraries must rely on vendors. Even if they can find a capable vendor and gather the necessary funds, a method that suggests how to locate subject headings using search results without ranking is unknown. The other problem with OPAC mining is that no subject heading is suggested when an input query is not included in OPAC's bibliographic information (e.g., book titles, author names, publisher names, etc.). This occasionally happens when input terms are relatively new.

We therefore investigate a method that suggests subject headings using "free" Web information sources instead of customizing OPACs. Our method suggests subject headings based on user queries by expanding queries using Web information sources. The advantages of our method include: (1) availability for any library without customizing OPACs, and (2) ability to suggest subject headings when a query string is not included in an OPAC's bibliographic information. In this research, we examine Wikipedia [2], Amazon [3], and Google [4] as free Web information sources for reasons stated in Section 3.

Below, in Section 2 we explain the subject headings and BSH4 subject headings [5] used in this research. Our method is described in Section 3, and experimental results are described in Section 4. We discuss our method's usefulness and related work in Section 5. The examples in this paper were translated from Japanese into English for publication.

2. BSH4 Subject Headings

Subject headings are controlled vocabularies in libraries sent to collections to help subject searches. The most famous subject heading is the Library of Congress Subject Headings (LCSH) [6]. In Japan, Basic Subject Headings (BSH) and National Diet Library Subject Headings (NDLSH) [7] are representative. In this research, we investigate a method to suggest BSH4, which is the latest edition of the most popular subject headings in Japan.

BSH4 has 7,847 subject headings and 2,873 entry terms. We treat these 10,720 terms as BSH4 subject headings, which feature "Used for (UF)," "Top Term (TT)," "Narrower Term (NT)," and "Related Term (RT),"as in a thesaurus. For example, the BSH4 subject heading "information retrieval" has "IR" as a UF, "information science" as a TT, "information science" as a BT, and "indexing" and "database" as NTs.

In Sections 3 and 4, we abbreviate "BSH4 Subject Headings" to "subject headings."

3. Method

3.1. Overview of our method

Our method is based on the vector space model and consists of three steps.

Step 1. Creating subject heading vectors. A subject heading is expanded using narrower terms to create a subject heading vector. Each subject heading vector is stored in a subject heading database.

Step 2. Creating a query vector. When a user inputs a query, it is expanded using information obtained from Wikipedia, Amazon, and Google, and its query vector is created.

Step 3. Calculating similarity. The similarity between the query vector and subject heading vectors is calculated using a cosine measure, based on a vector space model. As a result, ten subject headings are suggested.

Step 1 is a preparation process for a subject heading database. Steps 2 and 3 are processed continuously when a user inputs a query.

Figure 1 shows an overview of our method. When a user inputs "Java," our method suggests ten subject headings including "computer programming" and "Internet."

In this paper, terms are defined as nouns consisting of two or more characters; indexing terms for creating both subject headings and query vectors are defined as nouns consisting of two or more characters extracted from subject headings. Nouns consisting of one character are deleted because they are not very useful for indexing terms in Japanese. Here we call this the "deleting one-character heuristic."

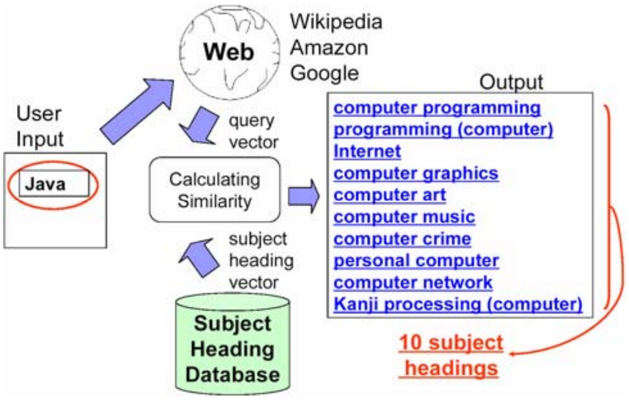

Figure 1. Overview of our method.

3.2. Creating subject headings vectors

A subject heading is expanded using narrower terms to create its subject heading vector, which is stored in a database.

For example, the subject heading "information retrieval" has an NT "indexing method" and "database." The "indexing method" has a NT "punch card." In this case, a subject heading vector for "information retrieval" becomes [information (1), retrieval (1), indexing (1), punch (1), card (1), database (1)], as shown in Figure 2. The number inside () is the frequency of each term. "Method" is removed due to the deleting one-character heuristic. In this way, all subject headings are converted to subject heading vectors.

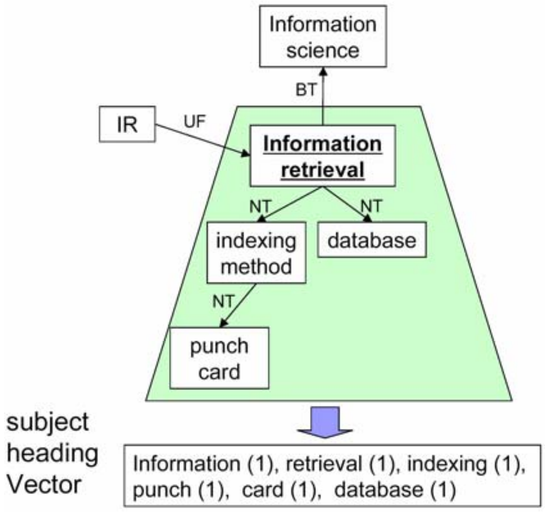

Figure 2. Creating a subject heading vector for subject heading "information retrieval".

3.3. Creating a query vector

3.3.1. Reasons for using Wikipedia, Amazon, and Google

It is common to add terms contained in databases in query expansion. However, this research's starting point is that "we cannot use terms contained in OPAC databases." We therefore investigate what kind of information is useful as a source of query expansion. We examined Wikipedia, Amazon, and Google for the following reasons.

Wikipedia is a free encyclopedia available on the Web. According to Guiles [8], Wikipedia approaches Britannica in terms of the accuracy of its science entries. We believe it is beneficial to use information obtained from Wikipedia to get relatively new terms related to user query definitions.

Amazon is one of the most popular shopping sites on the Web. In particular, Amazon book search is famous for the amount and quality of its database. In Amazon's book databases, category data are assigned to each book as subject information in OPACs. In addition, Amazon's search results can be ordered using amount of sales. We therefore thought we could get similar terms to those obtained in OPAC mining using Amazon.

Google is one of the most frequently used Web search engines. Its ranking of search results has a good reputation. When information is unavailable from Wikipedia or Amazon, Google functions as a complementary source for query expansion. We thus believe that we can get good terms for query expansion by mixing terms extracted from Wikipedia, Amazon, and Google.

The procedure for creating a query vector (Step 2) is comprised of the following four steps.

Step 2.1. Getting texts from Wikipedia, Amazon, and Google. Texts from Wikipedia, Amazon, and Google are searched and obtained.

Step 2.2. Extracting terms from each text. Terms are extracted from each text obtained in Step 2.1 and weighted.

Step 2.3. Merging terms. Terms are merged to create one vector.

Step 2.4. Filtering terms. Indexing terms are extracted from the above terms to create a query vector.

Figure 3 displays an example of creating a query vector when the user query "Java" was input on July 10th, 2005. We explain this example in the following sections.

3.3.2. Getting texts from Wikipedia, Amazon, and Google

First, we discuss how to handle multi-sense words in this research. For example, when a user searches for "Java" in Wikipedia, the first result is "Java (Indonesian island)," the second result is "Java (coffee)," the third result is "Java (curry)", the fourth result is "Java (programming language)," and so on. In contrast, most of the top 100 results in both Amazon and Google are related to "Java (programming language)." Amazon and Google search results are ordered by reputable ranking algorithms. To obtain good terms for query expansion, we must concentrate on one sense to avoid confusing various terms. Assuming that Google's search results help define one popular sense, we use the Google site search to search Wikipedia. In this case, the defined sense becomes "Java (programming language)". In this research, we get one text from Wikipedia, three from Amazon, and five from Google.

Our method first searches Wikipedia using the query through its top page. If there is one result, then our method gets it; otherwise, it searches the Google site search for Wikipedia. If there are results, then it gets the first result and then searches Amazon (actually, AWS [9], see Section 3.3.3), and if there are results, then it gets the first to third results. Finally it searches Google (actually, Google Web APIs [10], see Section 3.3.3); if there are results, then it gets the first to fifth results.

3.3.3. Extracting terms from each text

Terms are extracted from each obtained text and weighted based on the frequency of each term. Each process is described below.

(1) Wikipedia

One Japanese Wikipedia article was obtained in Step 2.1. The procedure for extracting terms from a Wikipedia article is constructed from three steps. First, parts unnecessary for navigation are omitted. Second, terms are extracted and weighted based on the frequency of the term. Third, terms contained in <b> tags (bold elements) and <a> tags (anchor elements) in the upper part of the articles are weighted more because they are assumed to be important.

In the example in Figure 3, the article “Java (programming language)” is obtained. After terms are extracted from the article and weighted, “java” and “oak” etc., extracted from <b> tags, and “object,” “orient,” “programming,” “language,” “sun,” “microsystem,” etc., extracted from <a> tags, are weighted additionally.

(2) Amazon

Amazon Web Service (AWS) offers Amazon data in XML format as a service for developers. We use AWS to get precise data about book titles and category data assigned to book titles. Our method searches AWS (only Japanese books) and gets three results. Among the bibliographic information of these three books, texts inside <BrowseNodes> tags (these contain category data) and book titles are extracted. Terms are extracted from these texts.

In the example in Figure 3, three XML data are obtained. These book titles are “Easy Java,” “Design and Implementation of Java: Learning from Source Code Reading," and "Java parallel processing programming: architecture and latest API." All are related to the Java programming language. "computer," "internet," "programming," "general," "java," etc. are extracted from < BrowseNodes > tags of the first book "Easy Java."

(3) Google

Google Web APIs is a service for developers that offers Google data. Our method searches Google (only Japanese pages) and gets five results. To reduce the processing time, texts are limited to 1,000 characters from which terms are extracted.

In the example of Figure 3, five Google search results are obtained. These titles are "Free Download of Java Software," "Sun Microsystems–Java Technology," "Java Language-Wikipedia," "What is Java?-definition: computer term dictionary e-word," and "Download Java 2 SDK, Standard Edition, v. 1.4.2 13 (J2SE)." All are related to the programming language sense of 'Java.' Terms are extracted from these five texts that are limited to 1,000 characters. "free," "download," "java," and "software," etc. are examples of terms extracted from the first result.

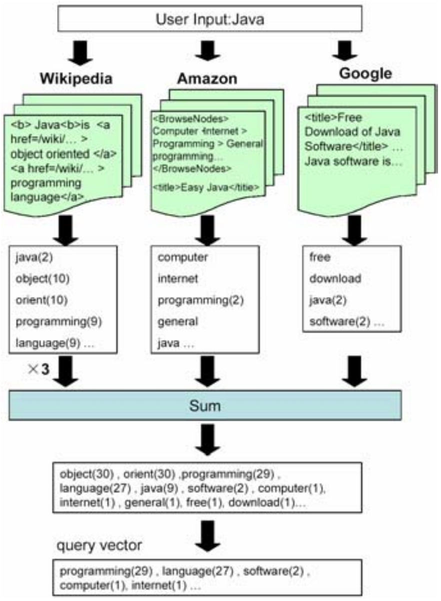
User Input:Java
Wikipedia
Amazon
Google
<b> Java<b>is <a href=/wiki/… > object oriented </a> <a href=/wiki/… > programming language</a>
<BrowseNodes> Computer ‹Internet > Programming > General programming… </BrowseNodes>
<title>Easy Java</title>
<title>Free Download of Java Software</title> … Java software is…
java(2)
object(10)
orient(10)
programming(9)
language(9) …
computer
internet
programming(2)
general
java …
free
download
java(2)
software(2) …
×3
Sum
object(30) , orient(30) ,programming(29) , language(27) , java(9) , software(2) , computer(1), internet(1) , general(1), free(1), download(1)…
query vector
programming(29) , language(27) , software(2) , computer(1), internet(1) …

Figure 3. Creating a query vector for user input "Java".

3.3.4. Merging terms

Terms are merged to create one vector. Terms extracted from Wikipedia are weighted three times as follows:

$$W(t) = 3Wi(t) + A(t) + G(t)$$

$W(t)$ is the weight of term t, $Wi(t)$ is the number of the frequency of terms extracted from Wikipedia, and $A(t)$ and $G(t)$ are those extracted from Amazon and Google, respectively. In our preliminary experiment, since information obtained from Wikipedia outperformed that from Amazon and Google, we set a higher weight for indexing terms obtained from Wikipedia.

In the example shown in Figure 3, the candidate query vector becomes [object (30), orient (30), programming (29), language (27), java (9), software (2)…].

3.3.5. Filtering terms

Indexing terms (see Section 3.1) are extracted from the query vector candidate to create a query vector.

In the example of Figure 3, since "object," and "oriented" are not included in indexing terms, they are deleted. When "Java" was input as a query, a query vector finally becomes [programming (29), language (27), software (2), computer (1), internet (1)…].

3.4. Calculating similarity

The similarity between the query vector and the subject heading vectors is calculated using a cosine measure, based on a vector space model [11].

Given a query vector: q and subject heading vector s_i, the similarity $\mathrm{sim}(q, s_i)$ is defined as follows:

$$\mathrm{sim}(q, s_i) = \frac{\sum_{j=1}^{t} w_{qj} w_{ij}}{\sqrt{\sum_{j=1}^{t} w_{qj}^2} \sqrt{\sum_{j=1}^{t} w_{ij}^2}},$$

where t is the number of terms, w_{qj} is the weight of t_j in query vector q, and w_{ij} is the weight of t_j in subject heading vector s_i.

As stated in Section 3.1, indexing terms to create both vectors are defined as nouns consisting of two or more characters extracted from subject headings.

3.5. Example

When "Java" was input as a query, the ten suggested subject headings were "computer programming," "programming (computer)," "Internet," "computer graphics," "computer art," "computer music," and "computer crime," "personal computer," "computer network," and "Kanji processing (computer)" on July 10th, 2005, as shown in Figure 1.

4. Experiment

Compared to OPAC mining, the main advantage of our research is that it suggests subject headings when the user query seldom or never appears in OPACs. Examples of such input include new words, jargon, ambiguous words, etc. However, it is difficult to use these words to obtain appropriate subjects to evaluate our method's usefulness. Since computer terms are relatively new and concrete and it is therefore easy to evaluate their meaning, we used them as input terms to obtain subjects easily.

e-words [12] is one of the most popular website dictionaries of computer terms in Japan. A list of the top 100 accessed computer terms [13] on 9th July, 2005 was used for the experiments.

4.1. Experiment 1

Experiment 1 evaluated the basic usefulness of our method for university students.

4.1.1. Method

The subjects were 41 undergraduate students. A questionnaire displayed one computer term and ten subject headings suggested by our method. Five questionnaires were allocated to each subject.

First, the subjects evaluated their familiarity with the term: 5: very well; 4: well; 3: neutral; 2: not very well; 1: not at all. We call this the "known" measure.

Next, they evaluated how much these suggested subject headings were related to the associated term: 3: related; 2: neutral; 1: unrelated. We call this the "related" measure. We judged data with related values of "3" as relevant to calculate "precision." The experiment was conducted on July 14th, 2005.

4.1.2. Results and discussion

The subjects rarely answered "1" or "2" as the known values for the term related to the values of the subject headings. Therefore, we analyzed the data whose known values were more than 2. As a result, related values attached to 31 terms were examined. These 31 terms are shown in Table 2 in Experiment 3.

Figure 4 shows the average related values at each rank of the ten suggested subject headings. The related value of the top rank (first) was the highest (2.34). The related values were over 2.00 except for the eighth rank.

Precision was 55% (21/38) at the top rank (first), 49% (56/114) in the top 3, and 41% (150/370) in the top 10. The above results suggest the basic usefulness of our method when we used computer terms.

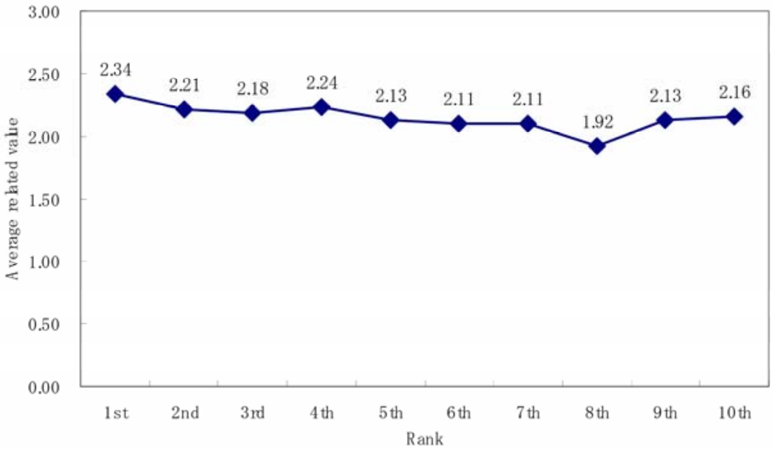

Figure 4. Average related values at first to tenth rank in Experiment 1.

4.2. Experiment 2

We examined the effectiveness of a combination of Wikipedia, Amazon, and Google compared to separately using each information source.

4.2.1. Method

We compared our method (d) with each source: (a) Wikipedia, (b) Amazon, and (c) Google. We used 31 terms defined by Experiment 1. Questionnaires were identical to Experiment 1.

After the subject, a computer science graduate student read the definitions of the 31 terms and evaluated how much the suggested subject headings were related to the terms, as in Experiment 1. She filled 31 (terms)* and 4 (conditions) questionnaires. The precision was calculated just as in Experiment 1.

The experiment was conducted on Dec 7th, 2005.

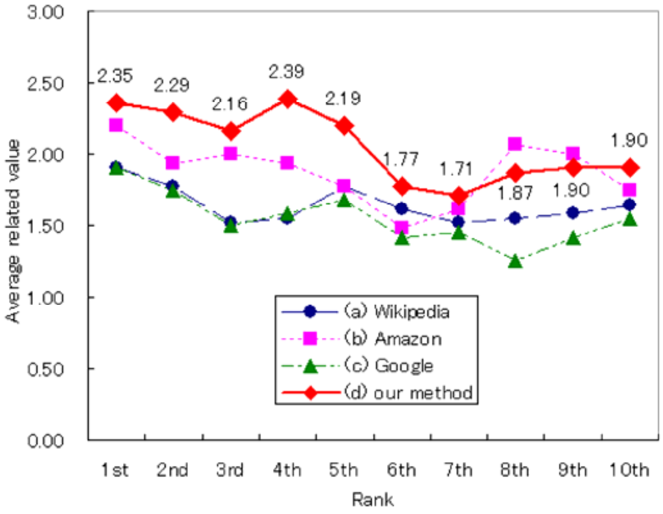

Figure 5. Average related values of four conditions at first to tenth ranks in Experiment 2.

4.2.2. Results and discussion

A one-way ANOVA was performed for average related values for each condition, and there was significant difference between each condition

($F(3, 36) = 12.86$, $p < .01$). In Fishers LSD test, there were significant differences between (d) our method and (a) Wikipedia, and (d) our method and (c) Google (both $p < .01$).

For the average related values of the top rank (first), our method (d) was highest (2.35) compared to (a) Wikipedia (1.90), (b) AWS (2.19), and (c) Google (1.90), as shown in Figure 5.

Table 1 outlines the precision results for each condition of the top 1, 3, and 10. Precision was 58% at the top rank, 55% at the top 3, and 43% at the top 10. Our method was the highest among all conditions.

We found a combination of Wikipedia, Amazon, and Google useful for query expansion to suggest subject headings.

Table 1. Precision of four conditions of top 1, 3, and 10 in Experiment 2

	Top 1	Top 3	Top 10
(a) Wikipedia	39% (12/31)	25% (23/93)	24% (73/310)
(b) Amazon	58% (18/31)	47% (44/93)	38% (119/310)
(c) Google	35% (11/31)	27% (25/93)	19% (60/310)
(d) our method	58% (18/31)	55% (51/93)	43% (132/310)

4.3. Experiment 3

Experiment 3 compared our method to OPAC mining.

4.3.1. Method

The subjects were six information science postgraduate students.

The first author searched Osaka City University's OPAC (keyword search for Japanese books) using 31 terms from Experiments 1 and 2. We checked ten search results for each term. For 10 of the 31 terms, the search obtained no subject heading, and for 4 of the 31 terms, only one subject heading was obtained. We deleted these 14 terms, and used the 17 terms shown in Table 2. The number of subject headings extracted from the top ten books for 17 terms ranged from 3 to 9. We extracted the top 3 subject headings using the frequency for each term. In this experiment, we treat the above method as "pseudo-OPAC mining" because the order of search results of Osaka City University's OPAC is alphabetized by book titles and does not use ranking as RedLightGreen.

A questionnaire displayed one term and three subject headings suggested by our method and pseudo-OPAC mining. 31 questionnaires were allocated to each subject, who evaluated how much the suggested subject headings were related to the terms, as in Experiments 1 and 2. We used the same method for calculating precision as in Experiments 1 and 2.

The experiment was conducted on Jan 18th, Apr 19th, and Apr 24th, 2006.

Table 2. 31 Terms and their results in Experiment 3

Computer term	Number of search results	Number of obtained books	Number of obtaind subject headings	Note
file-swapping software	0	0	0	10 terms (no subject heading is obtained.)
application software	0	0	0	
WMA	0	0	0	
Window x64 Edition	0	0	0	
MPEG4	0	0	0	
MP3	0	0	0	
DVD-RW	0	0	0	
ADSL	0	0	0	
proxy	1	1	0	
trojan horse	1	1	0	
WinMX	1	1	1	4 terms (only one subject heading is obtained.)
install	3	3	1	
CPU	4	4	1	
format	6	6	1	
HTTP	4	3	3	17 terms (more than one subject heading is obtained.)
JPEG	7	7	3	
CGI	11	10	7	
domain	13	10	8	
blog	21	10	7	
HTML	31	10	4	
archive	34	10	4	
ISDN	38	10	5	
Linux	47	10	3	
WWW	58	10	6	
cookie	92	10	9	
protocol	120	10	8	
OS	145	10	4	
Java	188	10	6	
DVD+RW	286	10	4	
Windows	356	10	6	
server	596	10	8	

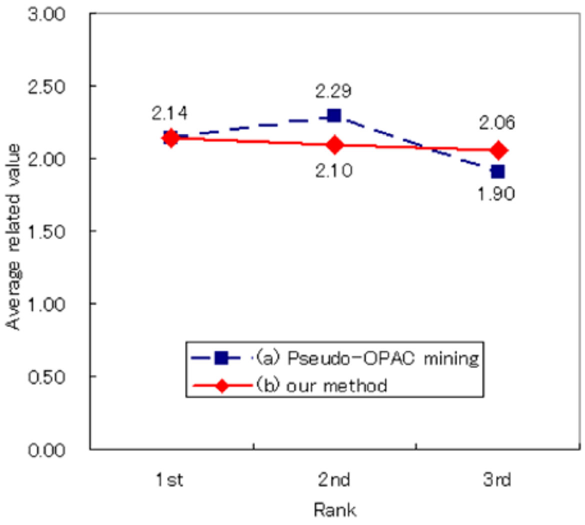

Figure 6. Average related values at first to third ranks in our method and pseudo-OPAC mining in Experiment 3.

4.3.2. Results and discussion

For 8 (26%) of 31 terms (e.g., file-swapping software, application software, etc.), there were no search results. For 2 terms (6%) (e.g., proxy, trojan horse), no subject heading was contained in any search results. In these cases, neither pseudo-OPAC mining nor OPAC mining could suggest subject headings.

For 4 terms (13%), pseudo-OPAC mining could only suggest one subject heading. In this case, our method outperforms pseudo-OPAC mining. We assume that our method might be better than OPAC mining.

For the remaining 17 terms, the results are as follows. Figure 6 shows the average related values at each rank of the three suggested subject headings for both our method and pseudo-OPAC mining. The average related value of the top rank was 2.14, the second rank was 2.10, and the third rank was 2.06 in our method. For pseudo-OPAC mining, the top rank was 2.14, the second rank was 2.29, and the third rank was 1.90. In precision, our method was 42% (64/153) and pseudo-OPAC mining was 42% (64/153). There was no significant difference between our method and pseudo-OPAC mining.

The above results show the following: (a) our method can suggest subject headings when pseudo-OPAC mining cannot; and (b) our method and pseudo-OPAC mining are nearly equal when we use computer terms. Thus our method can serve as an alternative to pseudo-OPAC mining. When we qualitatively analyze the results, a merit of our method is the newness of the suggested terms. In pseudo-OPAC mining, the suggested subject headings tended to include older terms at a glance. Subject headings once assigned to collection in OPACs are seldom deleted, so old subject headings remain forever. Old books have old subject headings, however new they may have been when the books were written.

5. Related Work and Discussion

5.1. Related work

The research described in this paper is a part of a project called Subject World [14], which visualizes such concepts as subject headings and classifications and enables users to explore these concepts and search OPACs. We explore methods for suggesting BSH4 and NDLSH subject headings, NDC9 [15] classifications, and so on when a pattern-matching algorithm fails. This is the first paper to describe our idea, and the three experiments that suggest BSH4 subject headings are described.

The objectives of the existing systems that suggest subject headings are divided into two. One helps users modify their queries. Our research is classified here. Another helps indexers who assign subject headings to collections. A typical example is to suggest MeSH subject headings in MEDLINE based on paper abstracts, which helps indexers assign appropriate subject headings for the abstracts [16, 17, 18].

Concerning methods, existing systems that suggest subject headings are based on pattern-matching or OPAC mining, as stated in Section 1. RedLightGreen is an example of OPAC mining.

Webcat Plus [19] displays terms included in book titles that are included in search results. It is identical to RedLightGreen in the sense that it uses OPAC mining, but different because it displays natural terms, not a controlled vocabulary like subject headings.

Many research results support the use of natural terms drawn from Web information sources instead of Data Mining. In addition, there is much research concerning automatic thesaurus construction using Web information sources (e.g., [20, 21]). Our method does not suggest natural terms, but uses them as sources for query expansion to suggest subject headings.

There are many query expansion studies for Web document retrieval using Web information sources (e.g., [22, 23]). We use Web information sources for query expansion to obtain subject headings to seek OPACs.

5.2. Discussion

We investigated a method to suggest BSH4 subject headings for subject searches in OPACs. Most existing systems use a pattern-matching algorithm. If it fails, no subject heading is displayed. Some research systems display subject headings using OAPC mining, which is based on the ranking of search results. However, few systems have adopted ranking of search results. How search results are ranked at an individual library is unknown. Moreover, due to budget shortages, it is difficult to implement OPAC mining even if it becomes clear how to rank terms in individual libraries. We pointed out two advantages of our method: (1) availability for any library without customizing OPACs, and (2) ability to suggest subject headings when a query string is not included in the OPAC's bibliographic information. Here we discuss the advantages and disadvantages of our method compared to OPAC mining.

The biggest advantage of our method is that it can suggest subject headings when OPAC mining cannot. This happens when user queries are not included in bibliographic information (mainly book titles) in OPACs. One typical example is when user queries consist of relatively new words. It takes several months from the birth of a neologism before it is included in book titles. It also takes a few more months from the publication of such a book to its registration in OPACs. In contrast, it may take just a few days before neologisms appear on Google and Wikipedia.

Next, the size of libraries, in other words, the number of collections, is different. In libraries with small collections, it becomes more difficult to suggest good subject headings because the number of search results is small. Our method has an advantage when the size of libraries is small. Third, since we need not rely on vendors to customize OPAC package software, our method's implementation is cheaper than OPAC mining, as stated in Section 1.

Fourth, the results of Experiment 3 showed that our method can suggest relatively new words better than OPAC mining. Since these words are more familiar than words suggested by OPAC mining, it may be useful to support end-user searching.

Finally, we will describe one example in which our method can suggest better subject headings than pattern-matching algorithms. When "orz" (a Japanese emoticon representing a kneeling or bowing person) is an input query, OPAC hits a German document that contains "ORZ" in its title. In contrast, our method suggested such subject headings as "modern term" and "current term."

There are some disadvantages in our method. First, the general precision of subject headings in our method is assumed to be worse than OPAC mining. In Experiment 3, the evaluation of our method nearly equals pseudo-OPAC mining. OPAC mining is assumed to be better than pseudo-OPAC mining.

Second, our method cannot deal with ambiguity, as stated in Section 2. If a user wants subject headings related to "the island of Java" in the example, our method fails.

Future work is listed below. First, we must improve our method. For example, we only weighted terms extracted from Wikipedia. We must examine weight terms extracted by Amazon and Google. We expanded user queries using Web information sources but did not expand subject headings. We need to expand subject headings using Web information sources and test the method. Second, we need to examine different kinds of data as user queries. Finally, we must examine the effectiveness of subject searches using suggested subject headings.

6. Conclusion

We proposed a method that suggests subject headings based on user queries when a pattern-patching algorithm fails to locate subject searches for OPACs. We combined information obtained from Wikipedia, Amazon, and Google for query expansion.

Our method has two advantages: (1) availability for any library without customizing OPACs, and (2) ability to suggest subject headings when a query string is not included in OPAC's bibliographic information.

Three experimental results using computer terms revealed the following. (1) Suggested subject headings were related to the input term. (2) Suggested subject headings were better when we used a mixture of Wikipedia, Amazon, and Google than when we just used one of them. (3) Our method can suggest subject headings when OPAC mining cannot.

We conclude that our method can serve as an alternative when pattern-matching algorithms fail.

References

[1] http://www.redlightgreen.com/

[2] http://ja.wikipedia.org/

[3] http://www.amazon.co.jp/

[4] http://www.google.co.jp/

[5] B.S.H. Basic Subject Headings 4th edition, the Committee on Subject Headings of the Japan Library Association, 1999.

[6] LCSH, http://authorities.loc.gov/

[7] NDLSH, ttp://www.ndl.go.jp/jp/library/data/ndl_ndlsh.html

[8] G. Guiles, Internet Encyclopedias Go Head to Head, Nature 438(15) (2005), 900-901.

[9] http://aws.amazon.com/

[10] http://www.google.com/apis/

[11] G. Salton and J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

[12] http://e-words.jp/

[13] http://e-words.jp/p/s-ranking.html

[14] H. Murakami, T. Hirata and K. Kita, Subject World: A System for Visualizing OPACs, Proceedings of PNC Annual Conference and Joint Meetings 2002, 237-240, 2002.

[15] Nippon Decimal Classification: N. D. C., Compiled by Mori-Kiyoshi, Newly revised 9th edition. Revised by the Committee of Classification, Japan Library Association, 1995.

[16] A. R. Aronson, O. Bodenreider, H. F. Chang, S. M. Humphrey, J. G. Mork, S. J. Nelson, T. C. Rindflesch and W. J. Wilbur, The NLM Indexing Initiative, Proceedings of the 2000 AMIMA Annual Fall Symposium, 17-21, 2000.

[17] K. H. Chen and C. T. Wu, Automatically Controlled-Vocabulary Indexing for Text Retrieval, Proceedings of the 12th Research on Computational Linguistics Conference, 171-185, 1999.

[18] K. Shin, S. Y. Han and F. G. Alexander, Balancing Manual and Automatic Indexing for Retrieval of Paper Abstracts, TSD-2004, 203-210, 2004.

[19] Webcat Plus, http://webcatplus.nii.ac.jp/

[20] K. Nakayama, T. Hara and S. Nishio, Wikipedia Mining to Construct a Thesaurus, IPSJ Journal 47(10) (2006), 2917-2928.

[21] S. Sato and Y. Sasaki, Automatic Collection of Related Terms from the Web, ACL-03 Companion Volume to the Proceedings of the Conference, 121-124, 2003.

[22] M. S. Khan and S. Khor, Enhanced web document retrieval using automatic query expansion, Journal of the American Society for Information Science and Technology 55(1) (2004), 29-40.

[23] S. C. Wang and Y. Tanaka, Topic-oriented query expansion for web search, Proceedings of the 15th International Conference on World Wide Web, 1029-1030, 2006.