FLEXIBLE QUERY MODIFIERS FOR RESEARCH SUPPORT SYSTEM PAPITS

Nobuhiro Fujimaki, Tadachika Ozono and Toramatsu Shintani
Department of Intelligence and Computer Science
Nagoya Institute of Technology
Gokiso, Showa-ku,
Nagoya, Aichi, 466-8555, Japan
email: {fujimaki,ozono,tora}@ics.nitech.ac.jp

ABSTRACT
In this paper, we describe flexible query modifiers for effective research information retrieval in the research support system Papits. Since the existing queries that only combine "AND", "OR" and selected keywords, the existing information retrieval systems can not retrieve flexibly, and users have to repeat requery. So we extend an existing query in such a way as to allow fewer requeries. For this purpose, we propose two kinds of query modifiers: semantic modifier and ranking modifier. The semantic modifier gives the role of the keyword in the query. The ranking modifier is ordering a result of retrieval. Users can create a flexible query to retrieve the relevance information by using the query modifier. Moreover, we propose a domain specified retrieval algorithm. We use this algorithm in order to retrieve by a query specified by the semantic modifier. In this paper we present the implementation and the evaluation of our method.

KEY WORDS
Information Retrieval, Thesaurus, Query Expansion

1. Introduction

The increasing number of digital documents has mandated development of knowledge management systems, that is, if users are to re-use information effectively. We implement the research support system Papits[1, 2] that is one of the knowledge management systems. Papits make it possible for users to share research information. Advanced information retrieval technology is required to realize knowledge management systems including Papits to re-use information effectively. But there is no methodology to represent a query flexibly in existing information retrieval systems. In the existing query description method, we use "AND" and "OR" to narrow down or to expand the result of the retrieval. However it is difficult to represent user's information needs by using only "AND" and "OR". For example, by using "AND", the result may be too narrowed down, and by using "OR", the result may be too expanded. So users need to repeat requery for finding relevant information.

In order to decrease requery, we improve existing query description method. One of the problems of the existing queries is that users can not specify the role of the keywords to retrieve required information. We propose query modifiers to specify the role of the keywords. They are semantic and ranking modifiers. The semantic modifier specifies the role of the keywords in the query, and it use for query expansion. The ranking modifier specifies the ranking method of the retrieval results. By using the query modifiers, users can explicitly specify the meaning of the keywords in the query. For example, if a keyword in the query is a multisense word, in our approach, users explicitly specify the meaning of the keyword. Therefore users can reflect their information needs in a query more flexibly than the existing query. Moreover, we propose a domain specified retrieval algorithm. The algorithm determines whether a document containing a keyword is a document relevant to a domain by using co-occurrence relationships of words in documents containing a domain keyword. Thereat we use co-occurrence based thesaurus[3].

The remainder of this paper is organized as follows: in Section 2, we describe the query modifier. In Section 3, we describe two modifiers of Papits. In Section 4, we describe our query expansion methods. In Section 5, we describe an application of our method and show its usefulness. Section 6, discusses related works. We conclude with a brief summary and future research directions.

2. Query Modifier

Existing queries are that a user inputs a query using keyword "AND", "OR". We use "AND" and "OR" to narrow down or to expand the result of the retrieval. However it is difficult for the user to represent user's information needs only using the keywords.

The query doesn't represent a concept of the word. For example, "Learning" is used in a domain of the artificial intelligence, the cognitive science, and the education. So the result of the retrieval includes documents that don't expect for the user. And more, the result of the retrieval is ranking by the system. If the ranking by the system and the ranking expected by the user are different, the user doesn't satisfy the result. Therefore we introduce semantic and ranking modifiers to extend existing query modifiers. If a user uses their modifiers, the user writes a query flexibly.

In addition, existing query modifier is too complex. So novices do not make a complex query. On the other hand, we propose to expand our query modifier with case-based query expansion method. The method expands the query using past case.

3. Semantic and Ranking Modifier

In this section, we describe two modifiers; semantic modifier, and ranking modifier. Semantic modifier gives semantics to a query. Ranking modifier gives a ranking ordering for result. They give the user a contain flexibility in writing queries.

3.1 Semantic Modifier

In our query modifier, we can elaborate by using the semantic modifiers. There are eight semantic modifiers show in Table 1. We explain four modifiers; domain, disambig, sim, and near.

- domain(k)

  This modifier is used to indicate a target domain for information retrieval. The target domain is represented by keyword k. The system retrieves information relevant to the target domain. If the user uses domain(k), the system suppresses to retrieval noise.

- disambig(k)

  This modifier is used to disambiguate of keyword k. Ambiguity of keyword k is that a concept is represented by different words. By eliminating the ambiguity of keyword k, the system retrieves documents including a concept represented by different words with k.

- sim(k)

  This modifier is used to find information with similar words to keyword k. Similar words can be selected by using similarity evaluation knowledge(SEK), which is automatically generate from existing text databases by using a data-mining algorithm[4]. "$R_{sim} : S \rightarrow \langle x, y \rangle$" is a SEK which means that a query constructed from keyword set S and keyword x, and an another query constructs from keyword set S and keyword y are able to exchange.

- near(k_i , k_j , n)

  This modifier is used to indicate the distance between keyword k_i and k_j . If two keywords occur 'near' each other, they are strongly related. Relations between keywords depend on information domain; therefore, this modifier indicates the information domain.

Nested semantic modifiers are also acceptable in our query modifier. Combinations of several semantic modifiers can represent user's information needs more appropriately.

3.2 Ranking Modifier

The system ranks the results of retrievals from information sources. The ranking mechanism scores the results of any evaluation method. Ranking modifiers indicate the evaluation method in the query. The ranking modifiers are as follows:

- citation

  This modifier evaluates documents that have many citations. It indicates the authority of the documents in the domain.

- author

  This modifier evaluates documents which are written by an author who wrote many documents. It indicates the main researchers in the domain.

- download

  This modifier evaluates download count.

- similarity

  This modifier evaluates the similarity between retrieved documents and documents gathered by users.

- similarity(time)

  This modifier evaluates similarity between retrieved documents and documents gathered by the user in recent. This modifier is used to gather documents in the domain of interest to the user.

- similarity(case)

  Papits has a mechanism to estimate user preference based on past retrieval cases. The documents that are discovered and selected by the user include user preference, and the documents that are not selected by the user do not include user preference. Papits makes a thesaurus from the documents of the user. Thesaurus thus represents user preference. The system calculates similarity between the user thesaurus and document thesauri Papits ranks by using these similarities. If a document thesaurus is more similar to the user thesaurus then the document is highly ranked, and if a user thesaurus is more similar to another user thesaurus then documents of the user are highly ranked.

4. Query Expansion using Co-occurrence Thesaurus

Query expansion has long been suggested as a technique for dealing with the fundamental issue of word mismatch

Table 1. Semantic Modifiers of Papits

k	retrieve documents including a keyword k.
domain(k)	target a domain of keyword k.
weight(k,n)	weight keyword k to n.
disambig(k)	eliminate ambiguous keyword k.
sim(k)	retrieve documents including words that are similar to keyword k
near(k_i,k_j,n)	retrieve documents co-occuring with keywords k_i and k_j within distance n.
and(k_i,k_j)	retrieve documents including keywords k_i and k_j.
or(k_i,k_j)	retrieve documents including keywords k_i or k_j.
not(k)	retrieve no documents including a keyword k.

in information retrieval. Query expansion is commonly used to add useful words to a query. The various query expansion methods can be classified into several types. We use the simplest method, that of use of co-occurrence data.

4.1 Building Co-occurrence Thesaurus

A recent work on automatic thesaurus construction uses term co-occurrence data[3]. Here, we define thesaurus being the Co-occurrence Thesaurus.

The basic idea of the co-occurrence thesaurus is that the context of a concept can be used to determine similarities among concepts. The context can be defined in a number of ways, as can concepts. The simplest definitions are that all words are concepts and that the context for a word is all the words that co-occur in documents with that word.

We use average conditional probability(ACP) as the similarity measure, which was used in the collocation map[3] for automatic thesaurus construction. ACP is the average value of two conditional probabilities $P(x|y)$ and $P(y|x)$.

$$cooccurrency(x,y) = \frac{P(x|y)+P(y|x)}{2}. \quad (1)$$

For example, suppose that co-occurrence terms are defined as terms occurring in the same document. Then, let N be the number of documents, df_x the term x of document frequency, and df_{xy} is the co-occurrence frequency of two terms, $P(x) = df_x/N$ and $P(xy) = df_{xy}/N$. Therefore the similarity of term x and term y is calculated as:

$$cooccurrency(x,y) = \frac{df_{xy}}{2}(\frac{1}{dfx}+\frac{1}{dfy}). \quad (2)$$

4.2 Expanding Query using Co-occurrence Thesaurus

In this section, we describe an algorithm to interpret the semantic modifier and ranking modifier. To evaluate the similarity of two thesauri, we use terms for the same concept in two thesauri. The evaluation is calculated using Equation 3.

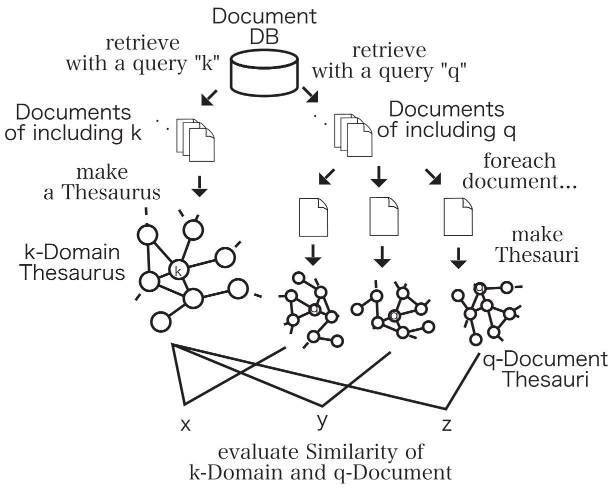


Figure 1. Algorithm

$$relation(t_1,t_2,x) = \frac{\sum_y C_{t_1}(x,y) * C_{t_2}(x,y)}{\sum_y C_{t_1}(x,y)^2 \sum_y C_{t_2}(x,y)^2} \quad (3)$$

where t_1 and t_2 are thesauri, x is a term of the target, y is a term in the thesauri, and $C_t(x,y)$ is coocurrency of x and y in thesaurus t. The term relation is calculated as the cosine of the term vector.

The similarity of two thesauri are calculated using Equation 4.

$$similarity(t_1,t_2) = \frac{1}{n}\sum_x relation(t_1,t_2,x) \quad (4)$$

where n is the number of terms in thesaurus t_1 or thesaurus t_2. The similarity of two thesauri is calculated as the average of the x's. We use this similarity to decide if a domain includes a document.

Figure 1 shows our thesaurus-based document ranking method. Our method uses a document database and a full-text search method. The algorithm is as follows:

(step 1) Retrieve documents with keyword k from the database The keyword k indicates the domain of the user's request.

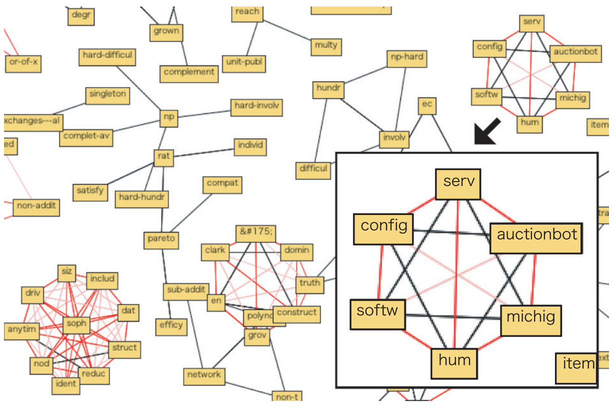


Figure 2. Auction-Domain Thesaurus

(step 2) Construct a domain thesaurus from the retrieval result. The domain thesaurus is calculated from documents including the keyword k. Separate each document into sentences, and separate the sentences into words. Calculate the co-occurrence of word combinations by using Equation 2, where df_x is sf_x, and df_{xy} is sf_{xy}. sf is sentence frequency. sf_x is the number of sentences including word x, and sf_{xy} is the number of sentences including words x and y. The thesaurus represented as a graph that has nodes and arcs.

(step 3) Retrieve documents in the database with keyword q. The keyword q directly indicate user's request.

(step 4) Construct document thesauri using the documents of the retrieval result. Separate the document into sentences, and separate the sentences into words. Calculate the co-occurrence of each word combination using Equation 2.

(step 5) Evaluate the similarity of the domain thesaurus and each document thesaurus, and compare the similarities to rank the documents.

If a document includes the domain-keyword k and the keyword q then the document satisfies the user's request. However even if the document does not include the domain-keyword k, the domain indicated by k. Our method also targets it may still be in documents including the keyword q.

Figure 2 shows a co-occurrence based thesaurus. In the figure, the co-occurence words are connected with the link. The thesaurus is made from a retrieval result with keyword "auction". For example, in the thesaurus, "auctionbot", "softw", "config", "serv", "hum" and "michig" are co-occuring (each words are stem; "softw" means "software").

4.3 Expanding Query using Cases

Papits expands keywords in a query inputted by users. To retrieve keywords A and B, or A and B and C after retrieving keyword A, the user can narrow the range of the result by using keyword B or C. Therefore we modified the query "and(A,B)" into "and(domain(A),B)". This expansion makes it possibe to retrieve documents including keyword B from the domain of A.

To retrieve keywords A and C after retrieving keyword A and B, the user can modify the query to eliminate ambiguity between B and C. Therefore we modified the query "and(A,C)" into "and(A,disambig(C))". This expansion makes be able to retrieve documents including ambiguity keyword other then B or C.

The system expands a query by using past cases of the retrieval. Frequent keywords (in past cases) indicate the domain. Therefore the keywords are completed by a domain-modifier.

These query expansions are represented by IF-THEN rules. They are independent of each other, so it is easy to add a new rule or delete a old rule. Their rules represented by the following formulas:

$$and(before(A), current(and(A,B))) \Rightarrow current(and(domain(A),B)) \qquad (5)$$

$$and(and(before(and(A,B)), current(and(A,C))), ambiguous(B,C)) \Rightarrow current(and(A,disambig(C))) \qquad (6)$$

$$and(frequency(A,case), current(A)) \Rightarrow current(domain(A)) \qquad (7)$$

To describe rules, we define these predicates: "current(k)" indicates that k is a current keyword. "before(k)" indicates that k is a previous keyword. "ambiguous(k_i ,k_j)" indicates that keywords k_i and k_j are ambiguous. "frequency(k,case)" indicates how frequently keyword k occurs in the case.

The expanded query is sent to the database retrieval mechanism.

5. Research Support System Papits

In this section, we describe the research support system Papits.

Figure 3 illustrates the retrieval process in Papits. A user writes query Q in our query modifier and inputs it into Papits. The query expansion module shown in figure 3 expands the query to reflect the meanings of semantic modifiers. The module also uses past cases to expand queries. Let Q' to be an expanded query of Q. The database retrieval module retrieves documents relevant to Q' from the document DB. The result of retrieval with Q' is set to R. The result ranking module ranks R with respect to ranking modifiers and past cases. The ranked result is set to R'. Ranked result R' is provide to the user. Finally, the system stores users' feedback in the casebase.

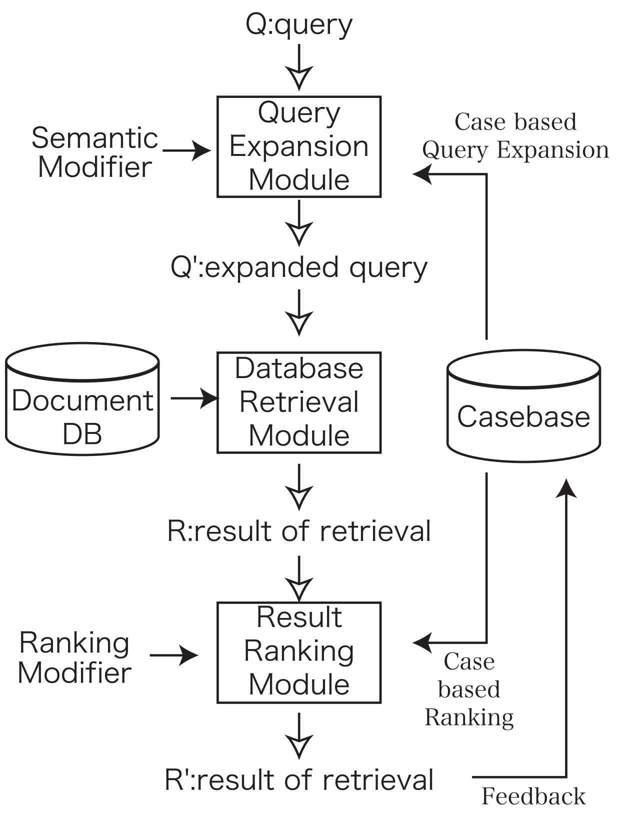


Figure 3. System Work Flow

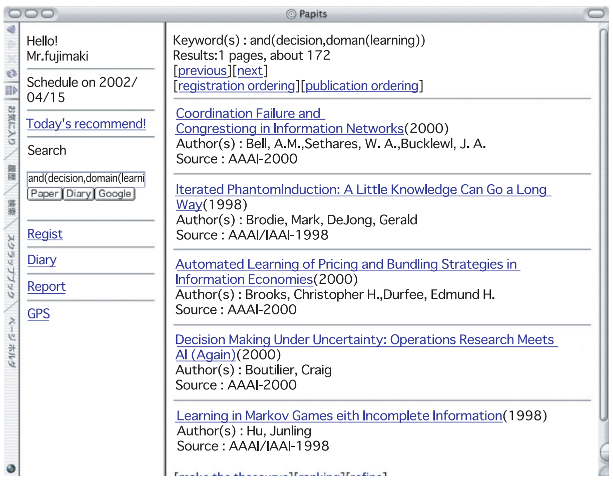
5.1 Implementation

Papits is implemented as a web application (by using WebObjects), and can be used with web browsers. Papits has several subsystems to manage research information, including a scheduler, a paper recommender, a paper retriever, a research diary, a research report, and a location manager. A user makes a query elaborates it by using semantic and ranking modifiers, and input it to the system. The paper retriever attempts to extend the query to improve the results.

The user inputs a query using semantic modifier and ranking modifier. And our query expansion module expands the input query. Expansion rules are used to expand the query. Rule adaptation is iterated if adaptable rules exist. When all rules have been adapted, the expansion has finished. Rules are adapted by pattern matching.

Documents are retrieved by using the extended query. In addition to searches for full text, the system retrieves documents to evaluate similarity between thesauri. The retrieval results are represented as a set of full text search results thesaurus-based similarity evaluation results. The results are in the form of a ranked list. Figure 4 is the ranked list of a result of retrieval with a keyword “and(decision,domain(learning))”. The result contains articles that contain “decision” and relevant “learning”. Ranking of results is indicated by a ranking modifier and is based on similarity of retrieval case.

Users can download documents they want to read. If they can not find necessary information, they retrieve


@ Papits

Hello!
Mr.fujimaki

Schedule on 2002/
04/15

Today's recommend!

Search

and(decision,domain(learni
Paper Diary Google

Regist
Diary
Report
GPS

Keyword(s) : and(decision,doman(learning))
Results:1 pages, about 172
[previous][next]
[registration ordering][publication ordering]

Coordination Failure and Congrestiong in Information Networks(2000)
Author(s) : Bell, A.M.,Sethares, W. A.,Bucklewl, J. A.
Source : AAAI-2000

Iterated PhantomInduction: A Little Knowledge Can Go a Long Way(1998)
Author(s) : Brodie, Mark, DeJong, Gerald
Source : AAAI/IAAI-1998

Automated Learning of Pricing and Bundling Strategies in Information Economies(2000)
Author(s) : Brooks, Christopher H.,Durfee, Edmund H.
Source : AAAI-2000

Decision Making Under Uncertainty: Operations Research Meets AI (Again)(2000)
Author(s) : Boutilier, Craig
Source : AAAI-2000

Learning in Markov Games eith Incomplete Information(1998)
Author(s) : Hu, Junling
Source : AAAI/IAAI-1998

Figure 4. Retrieval result

with other query again. If the user has downloaded the document, the user’s information needs have been satisfied. If the user makes another query, his or her information needs have not been satisfied and the system failed to support the retrieval. User’s operations are stored in a casebase and are used for future retrieval.

5.2 Evaluation

To evaluate this implementation of the system, we checked the behavior of the system as follows: A user input the query “and(decision, domain(learning))”. The result of retrieval with the query was articles that contain “decision” and relevant to “learning”. Therefore the user could retrieve not only articles that contain “learning”, but also relevance articles that do not contain “learning”. Moreover the system is filtering articles that contain “decision”, but do not relevant “learning” (for example, articles of “negotiation”). The other case, when a user input the query, “and(agent, disambig(auction))”, the system found that “agent” represents a domain in the casebase. Dependings on how “disambig(auction)” is used the number of retrieval results may or may not increase. The semantic modifier “domain” is applied to “agent”. To expand the query into “and(domain(agent), disambig(auction))” in order to narrow down the results. Therefore the system saved the trouble of the user having to narrow down the query by requerying.

6. Related Work

Making a query that represents user’s thought is difficult for novices. Several researchers[5, 6, 7, 8] attempt to build support systems for such users. Sakai proposed a method for generating the second keyword from the results of the first query to improve the results of the second query[5]. Kawahara discovered rules for query expansion by studying a data mining approach[6]. Sunayama proposed a visual interface to build a query[7]. For estimating user model, Suzuki generates a pattern of queries from the system log and uses this pattern to make retrieval more effective.

Several method of query expansion have been reported[9, 10]. Xu compare the effectiveness of global document analysis with local document analysis. And Xu reported that local document analysis is more effective than global document analysis[9]. Qiu proposed a probabilistic query expansion model that is based on a similarity thesaurus. The thesaurus is constructed automatically, and their method has the advantage that it is fully automatic and can be used in the first run of an IR system[10].

Approaches to constructing the thesaurus have been reported [3, 11]. Jing reported that is feasible to construct a useful collection-dependent association thesauri automatically without relevance judgments[11].

7. Conclusions

In this paper, we described a method to retrieve flexibly for the research support system Papits, and proposed new query modifiers. In the Papits, the queries combine semantic modifiers and ranking modifiers in the query modifiers. For example, this lets us indicate the domain of a retrieval target, eliminate the ambiguousness of a query, retrieve similar documents are directly queried, etc. With this new query modifier, the user is able to write a flexible query, but the modifier is complex.

To solve this problem, the query can be expanded into one that better represents the user's information needs by using past retrieval cases. We used expansion rules, and pattern matching to adapt the rules.

We applied the co-occurrence-based thesaurus to the domain specified retrieval. We treated the retrieval results with a domain specified keyword as the target domain. We used the similarity of the thesaurus made from the result to the thesaurus made from a document.

We are currently investigating ways of making co-occurrence thesauri more quickly. Making the thesaurus uses many resources and delays the response time of the system. We are also investigating ways of making the query expansion rules more effective. Because our query expansion method depends on the expansion rules, we should try to acquire new expansion rules.

References

[1] Shoji Goto, Tadachika Ozono and Toramatsu Shintani,A Method for Information Source Selection using Thesaurus for Distributed Information Retrieval, *In the Proceedings of the Pacific Asian Conference on Intelligent Systems 2001 (PAIS2001)*, Soul, Korea, 2001, 272-277.

[2] Tadachika Ozono, Shoji Goto, Nobuhiro Fujimaki, Toramatsu Shintani, P2P based Knowledge Source Discovery on Research Support System Papits,*The First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, Bologna, Italy, 2002, 49-50.

[3] Young C. Park, Young S. Han and Key S. Choi,Automatic thesaurus construction using Bayesian networks, *Proc. of the International Conference on information and knowledge management*, Maryland, USA, 1995, 212-217.

[4] Mineko Ohira, Tadachika Ozono and Toramatsu Shintani, Implementing A Recipe Search System, *In the Proceedings of the Sixty Second Annual Conference of Information Processing Society of Japan (IPSJ) (3)*,Tokyo, Japan, 2001, 129-130(in Japanese).

[5] Hiroyuki Sakai, Kiyonori Ohtake and Shigeru Masuyama, A Retrieval Support System by Suggesting Terms to a User, *In Proceedings of The Seventh Annual Meeting of The Association for Natural Language Processing(NLP)*, Tokyo, Japan, 2001, 185-188(in Japanese).

[6] Minoru kawahara, Hiroyuki Kawano and Toshiharu Hasegawa, Data Mining Technologies for Bibliographic Navigation System, *Journal of Information Processing Society of Japan(IPSJ)*, 39(4), 1998, 878-887(in Japanese).

[7] Wataru Sunayama, Yukio Ohsawa and Masahiko Yachida, A Search Interface with Supplying Search Keywords by Using Stracture of User Interst, *Journal of Japanese Society for Artificial Intelligence(JSAI)*, 15(6), 2000, 1117-1124(in Japanese).

[8] Shunsuke Suzuki and Hayato Yamana, Search Pattern Modeling based on its Search Interval, *IPSJ SIG Notes, 2002-DD-32*, Tokyo, Japan, 2002, 103-110(in Japanese).

[9] Jinxi Xu and W. Bruce Croft, Query Expansion Using Local and Global Document Analysis, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96*, Zurich, Switzerland, 1996, 4-11.

[10] Yonggang Qiu and Hans-Peter Frei, Concept-based query expansion, *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, Pittsburgh, USA, 1993, 160-169.

[11] Y. Jing and W. Bruce Croft, An association thesaurus for information retrieval, *Proceedings of RIAO-94, 4th International Conference "Recherched'Information Assistee par Ordinateur"*, New York, USA, 1994, 146-160.