Knowledge Navigation on Visualizing Complementary Documents

Naohiro Matsumura[1,3], Yukio Ohsawa[2,3], and Mitsuru Ishizuka[1]

[1] Graduate School of Engineering, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan
{matumura, ishizuka}@miv.t.u-tokyo.ac.jp
[2] Graduate School of Systems Management, University of Tsukuba,
3-29-1 Otsuka, Bunkyo-ku, Tokyo, 112-0012 Japan
osawa@gssm.otsuka.tsukuba.ac.jp
[3] Japanese Science and Technology Corporation,
2-2-11 Tsutsujigaoka, Miyagino-ku, Sendai, Miyagi, 983-0852 Japan

Abstract. It is an up-to-date challenge to get answers for novel questions which nobody has ever considered. Such a question is too rare to be satisfied with a past single document. In this paper, we propose a new framework of knowledge navigation by graphically providing with multiple documents relevant to a user's question. Our implemented system named MACLOD generates several navigational plans, each forming a complementary document-set, not a single document, for navigating a user to understanding a novel question. The obtained plans are mapped into a 2-dimensional interface where documents in each obtained document-set are connected with links in order to support user selecting a plan smoothly. In experiments, the method obtained satisfactory answers to user's unique questions.

1 Introduction

It is an up-to-date challenge to answer a user's novel question nobody has ever asked. However, such a question is too new to be satisfied with a past single document, and the required knowledge for understanding the documents relevant to a user's question depends on his background[4]. In our previous work[3], we proposed a novel information retrieval method named *combination retrieval* for creating novel knowledge by combining complementary documents. Here, a complementary set of documents is composed of documents, and the combination of which supplies a satisfactory information. This idea is based on the principle that combining ideas can trigger the creation of new ideas[1, 2]. Throughout the discussions of the work, we verified the fact that reading multiple complementary documents generates the synergy effects which help us acquire novel knowledge.

In this paper, we propose a new framework of knowledge navigation, i.e., supply a user with new knowledge, for satisfying the information request of a user by visualizing complementary documents. Our implemented system named

MACLOD(Map of Complementary Links of Documents) generates several navigational plans, each formed by a document-set for navigating a user to understand a novel question, by making use of the combination retrieval[3]. The obtained plans are mapped into a 2-dimensional interface where documents in each document-set are connected with links in order to support user selecting complementary documents smoothly.

The remainder of this paper goes as follows: In Section 2, the meaning of our approach is shown by comparison with previous knowledge navigation methods. The mechanism of combination retrieval is described in Section 3, and the mechanism of MACLOD implemented here is described in Section 4. We show the experiments and the results in Section 5, showing the performance of MACLOD for medical counseling question-answer documents.

2 Previous Methods for Knowledge Navigation

The vision of knowledge navigation was shown by John Sculley(Then the president of Apple Computer Inc.) where electronic secretary in a computer named Knowledge Navigator managed various tasks on behalf of users, e.g., manage schedules. The concept inspired us. However, it is still difficult to realize the Knowledge Navigator because of the complexity of real secretary's tasks.

A knowledge navigation system is a piece of software which answers a user's question. The question maybe entered as a word-set query $\{alcohol, liver, cancer\}$ or a sentence query "Does alcohol cause a liver cancer ?" An intelligent answer to this question may be "No, alcohol does not cause liver cancer directly. You may be confused of liver cancer and other liver damages from alcohol. Alcohol causes cancer in other tissues." For giving such an answer, the system should have medical knowledge relevant to user's query, and infer on the knowledge for answering the question. However, it is not realistic to implement such knowledge wide enough to be applied to unique user interests.

Another approach for navigating knowledge is to retrieve ready-made documents relevant to the current query, from a prepared document collection. In this way, we can skip the process of knowledge acquisition and implementation, because man-made documents represent the complex human knowledge directly. A search engines for a word-set query entered by the user may be the simplest realization of this approach. However, we already know that existing information retrieval methods trying to answer a query by ONE of the output documents could not satisfy novel interests in Section 1.

3 The Process of Combination Retrieval

Combination retrieval[3] is a method for selecting meaningful documents which, as a set, serve a good (readable and satisfactory) answer to the user. In this section, we review the algorithm of the combination retrieval.

3.1 The Outline of the Process

The process of combination retrieval is as follows:

The Process of Combination Retrieval
Step 1) Accept user's query Q_g.
Step 2) Obtain G, a word-set representing the goal user wants to understand, from Q_g ($G = Q_g$ if Q_g is given simply as a word-set).
Step 3) Make knowledge-base Σ for the abduction of Step 4). For each document D_x in the document-collection C_{doc}, a Horn clause is made as to describe the condition (words needed to be understood for reading D_x) and the effect (words to be subsequently understood by reading D_x).
Step 4) Obtain h, the optimal hypothesis-set which derives G if combined with Σ, by cost-based abduction (detailed later). h obtained here represents the union of following information, of the least size of K.
S**:** The document-set the user should read.
K**:** The keyword-set the user should understand for reading the documents in S.
Step 5) Show the documents in S to the user.

The intuitive meaning of employing the abductive inference is to obtain the conditions for understanding user's goal G. Here, conditions include the documents to read (S) for understanding G, and necessary knowledge (K) for reading those documents. That is, S means the combination of documents to be presented to the user.

3.2 The Details of Combination Retrieval's Process

In preparation, collection C_{doc} of existing human-made documents is stored. Key, the set of keyword-candidates in the documents in C_{doc}, i.e. word-set which is the union of extracted keywords from all the documents in C_{doc}, is obtained and fixed. Here, words are stemmed as in [5] and stop words ("does", "is", "a"...) are deleted, and then a constant number of words of the highest TFIDF values [6] (using C_{doc} as the corpus for computing document frequencies of words) are extracted as keywords from each document in C_{doc}. Next, let us go into the details of each step in 3.1.
Step 1) to 2) Make goal G from user's query Q_g: Goal G is defined as the set of words in $Q_g \cap Key$, i.e., keywords in the user's query. For example, "does alcohol make me warm?" and query $\{alcohol,\ warm\}$ are both put into the same goal $\{alcohol,\ warm\}$, if C_{doc} is a set of past question-answer pairs of a medical counselor which do not have "does", "make", "me", "warm", "in", "a", or "day" in Key (some are deleted as stop words).
Step 3) Make Horn clauses from documents: For the abductive inference in Step 4) of Subsection 3.1, knowledge-base Σ is formed of *Horn clauses*. A Horn clause is a clause as in Eq.(1), which means that y becomes true under the condition that all $x_1, x_2, \cdots x_n$ are true, where variables $x_1, x_2, \cdots x_n$ and y

are atoms each of which corresponds to an event occurrence. A Horn clause can describe causes $(x_1, x_2, \cdots, x_n)$ and their effect (y) simply.

$$y :- x_1, x_2, \cdots, x_n. \tag{1}$$

In combination retrieval, the Horn clause for document D_x describes the cause (reading D_x with enough vocabulary knowledge) and the effect (acquiring new knowledge from D_x) of reading D_x, as:

$$\alpha :- \beta_1, \beta_2, \cdots, \beta_{mx}, D_x. \tag{2}$$

Here, α is the *effect term* of D_x, which is a term (a word or a phrase) one can understand by reading document D_x. $\beta_1,\ \beta_2 \cdots \beta_{mx}$ are the *conditional terms* of D_x, which should be understood for reading and understanding D_x. That is, one who knows words $\beta_1,\ \beta_2 \cdots \beta_{mx}$ and reads D_x on this knowledge is supposed to acquire knowledge about α.

The method for taking the effect and the conditional terms from D_x is straight-forward. First, the effect terms $\alpha, \alpha_2, \cdots$ are obtained as terms in $G \cap$ (*the keywords of* D_x). This means that the effect of D_x is expected on the user's interest G, rather than by the intension of the author of D_x. For example, a document about cancer symptoms may work as a description of the demerit of smoking, if the reader is a heavy smoker. Focusing the consideration onto user's goal in this way also speeds up the response of combination retrieval as in Subsection 5.1.

Then, the keywords of D_x other than the effect terms above form the conditional terms $\beta_1,\ \beta_2,\ \cdots \beta_{mx}$. As a result, Horn clauses are obtained as

$$\begin{aligned} \alpha_1 &:- \beta_1, \beta_2, \cdots \beta_{mx}, D_x, \\ \alpha_2 &:- \beta_1, \beta_2, \cdots \beta_{mx}, D_x, \\ &\vdots \end{aligned} \tag{3}$$

meaning that one knowing $\beta_1,\ \beta_2, \cdots \beta_{mx}$ can read D_x and understand all the effect terms $\alpha_1,\ \alpha_2, \cdots$ by reading D_x.

Step 4) Cost based abduction for obtaining the documents to read: We employ the *cost based abduction* (CBA, hereafter)[7], an inference framework for obtaining solution h of the least $|K|$ in Subsection 3.1. In CBA, the causes of a given effect G is explained. Formally, CBA is described as extracting a minimal hypothesis-set h from a given set H of candidate hypotheses, so that h derives G using knowledge Σ. That is, h satisfies Eq.(4) under Eq.(5) and Eq.(6). We deal with Σ composed of causal rules, expressed in Horn clauses mentioned above.

$$Minimize\ cost(h),\ under\ that: \tag{4}$$

$$h \subset H, \tag{5}$$

$$h \cup \Sigma \vdash G, \tag{6}$$

Eq.(4) represents the selection of h to be minimal, i.e., of the lowest-cost hypothesis-set $h(\subset H)$, where cost denoted by $cost(h)$ is the sum of the *weights* of hypotheses in h. The weights of hypotheses in H, the candidates of elements of solution h, are initially given. Generally speaking, the weight-values of hypotheses are closely related to the semantics in the problem to which CBA is applied, as exemplified in [8]. In combination retrieval, weights are given differently to the two types of hypotheses in H:

Type 1: Hypothesis that user reads a document in C_{doc}
Type 2: Hypothesis that user knows (have learned) a conditional term in Key

In giving weights to hypotheses, we considered that user should be able to understand the output documents in S, with learning only a small set K of keywords from external knowledge other than C_{doc}. This is reflected to minimizing $|K|$, the size of K. That is, the weights of hypotheses of Type 2 are fixed to 1 and ones of Type 1 are fixed to 0, and the content of h is $S \cup K$. It might be good to give values between 0 and 1 to hypotheses of Type 2, each value representing the difficulty of learning each term. However, we do not know how each word is easy to learn for the user from outside of C_{doc}. Further, it might seem to be necessary to give positive weights to hypotheses of Type 1, each value representing the cost of reading each document. However, this necessity can be discounted because we gave mx in Eq. 3 to be proportional to the length of D_x. That is, the user's cost (effort) for reading a document is implied by the number of meaningful keywords s/he should read in the document. If we sum the heterogeneous difficulties, i.e., of reading documents and of learning words, the meaning of the solution cost would become rather confusing.

3.3 An Example of Combination Retrieval's Execution

For example, the combination retrieval runs as follows.

Step 1) Q_g = "Does alcohol cause a liver cancer ?"
Step 2) G is obtained from Q_g as $\{alcohol,\ liver,\ cancer\}$.
Step 3) From C_{doc}, documents D_1,D_2, and D_3 are taken, each including terms in G, and put into Horn clauses as:

$$alcohol :- cirrhosis, cell, disease, D_1.$$
$$liver :- cirrhosis, cell, disease, D_1.$$
$$alcohol :- marijuana, drug, health, D_2.$$
$$liver :- marijuana, drug, health, D_2.$$
$$alcohol :- cell, disease, organ, D_3.$$
$$cancer :- cell, disease, organ, D_3.$$

Hypothesis-set H is formed of the conditional parts of D_1, D_2 and D_3 of Type 1 each weighted 0, and "cirrhosis," "cell," "disease," "marijuana," "drug," "health,"and "organ" of Type 2 each weighted 1.

Step 4) h is obtained as $S \cup K$, where

$$S = \{\ D_1,\ D_3\}\ and$$
$$K = \{cirrhosis,\ cell,\ disease,\ organ\},$$

meaning that user should understand "cirrhosis", "cell", "disease" and "organ" for reading D_1 and D_3, served as the answer to Q_g. This solution is selected because $cost(h)$ (i.e. $|K|$) takes the values of 4, less than 6 of the only alternative feasible solution, i.e. $\{marijuana,\ drug,\ health,\ cell,\ disease,\ organ\}$ plus $\{D_2,\ D_3\}$.

Step 5) The user now reads the two documents presented as:

D_1 **(**including *alcohol* and *liver***)** stating that alcohol alters the liver function by changing liver cells into cirrhosis.

D_3 **(**including *alcohol* and *cancer***)** showing the causes of cancer in various organs, including a lot of alcohol. This document recommends drinkers to limit to one ounce of pure alcohol per day.

As a result, the subject learns that s/he should limit drinking alcohol to keep liver healthy and avoid cancer, and also came to know that other tissues than liver get cancer from alcohol.

Thus, user can understand the answer by learning a small number of words from outside of C_{doc}, as we aimed in employing CBA. More importantly than this major effect of combination retrieval, a by-product is that the common hypotheses between D_1 and D_3, i.e., $\{cell,\ disease\}$ of Type 2 are discovered as the context of user's interest underlying the entered query. This effect is due to CBA which obtains the smallest number of involved contexts, for explaining the goal (i.e. answering the query), as solution hypotheses. Presenting such a novel and meaningful context to the user induces the user to creating new knowledge [9], to satisfy his/her novel interest.

4 MACLOD: Map of Complementary Links of Documents

In the combination retrieval, a user was imposed on two types of tasks that reading a obtained document-set and understanding the conditional terms of the document-set. However, this tasks are not always easy for a user since the background knowledge of a user is different from individuals. For taking such already existing knowledge of a user into consideration when generating the document-set for reading, we propose a new framework to navigate a user by graphically providing with multiple documents which are constructed of some document-set each giving an answer to his/her interest. The implemented system named *MACLOD* (MAp of Complementary Links Of Documents) visualizes complementary documents obtained by iterating the combination retrieval. The process of MACLOD is as follows:

The Process of MACLOD

Phase1. Obtain a plan(document-set S) for user's query Q_g along the procedure of combination retrieval in Section 3. The documents obtained in S are in the complementary relations, and realize a coherent explanation for Q_g. The documents are merged into a satisfactory answer for Q_g in the user's mind[3].

Phase2. Iterate Phase1 to obtain N sets of plans by adding inconsistency conditions into knowledge-base Σ for avoiding already obtained plans. The inconsistency condition to be considered in a certain cycle of Phase1, is described as

$$inc :- D_{x1}, D_{x2}, \cdots, D_{xn}, \tag{7}$$

where $D_{x1}, D_{x2}, \cdots, D_{xn}$ are the documents obtained in one of the plans, already obtained in a cycle of previous Phase1. In addition, a document included in S three times, in previous series of Phase1 already, is forced not to be included for getting a new plan. This inconsistency condition, also added into knowledge-base Σ, is described as

$$inc :- D_{x1}. \tag{8}$$

Where D_{x1} is a document included in S already more than three times. The cycles of **Phase1** continues until the number of iterations reaches N. Here, we empirically set N as 10.

Phase3. MACLOD outputs a 2-dimensional interface in which obtained plans during above iterations are mapped. In the 2-dimensional interface, documents in a plan obtained by one cycle at Phase1 are connected with links each other in order to support user selects appropriate documents.

Phase4. The user goes on reading documents along the links in the 2-dimensional interface until s/he understands or gives up understanding Q_g.

In combination retrieval, user must understand the keyword-set K for reading the document S. However, this matter is not always easy for the user since human thoughts are individually different. MACLOD overcomes this weak spot by preparing several plans and pathways.

5 Experimental Evaluations of MACLOD

5.1 The Experimental Conditions

MACLOD is implemented in a Celeron 500MHz machine with 320MB memory. Although CBA is time-consuming because of its NP-completeness, most answers in the experiments were returned within ten seconds from the entry of query by high-speed abduction as in [12]. Queries from users included 4 or less terms in Key, due to which the response time was below 10 sec. This quick response comes also from the goal-oriented construction of Horn clauses shown in Subsection 3.2. The document-collection C_{doc} of MACLOD is 1808 question-answer pairs of *Alice*, a health care question answering service on WWW (`http://www.alice.columbia.edu`). The small number as 1808 documents is a suitable condition for evaluating MACLOD for a sparse document-collection which is insufficient for answering novel queries.

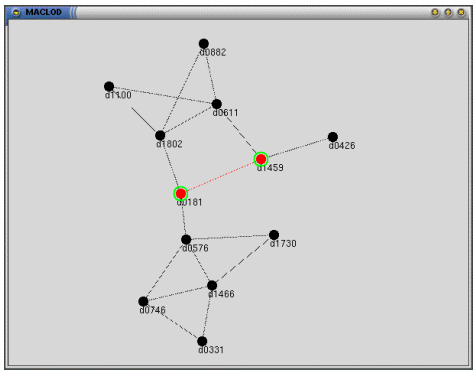

Fig. 1. A 2-dimensional interface of MACLOD. Documents are shown as nodes, and complementary documents are connected with links.

5.2 An Example of MACLOD's Execution

When a user entered a query in a word-set or a sentence, MACLOD obtained ten plans(document-sets) in Table 1 and showed a 2-dimensional output in Figure 5.2. In this case, input $\{alcohol,\ fat,\ calorie\}$ was entered as query Q_g for knowing if the calorie of alcohol changes into fat.

Table 1. The top 10 plans for the input query {alcohol, fat, calorie}.

Ranking	*Plan(document-set)*	*Cost*
1	$d1459$, $d0181$	25
2	$d1459$, $d0611$	26
3	$d1459$, $d0426$	27
4	$d1802$, $d0181$	27
5	$d0576$, $d0181$	27
6	$d1802$, $d0882$, $d0611$	39
7	$d1802$, $d1100$, $d0611$	39
8	$d0746$, $d0576$, $d1466$	39
9	$d1730$, $d0576$, $d1466$	39
10	$d0746$, $d0331$, $d1466$	41

The process of understanding the user's interest(shown as Q_g) begins by reading a document-set $d1459$ and $d0181$ (double-circle nodes in Figure 5.2), a top ranked plan of MACLOD. The summaries of them are as follows:

$d1459$ **(including** fat **and** $calorie$**)** stating that if the calorie comes short, the protein is burned into energy. The lack of protein delays the recovery of distress, or weakens the resistence to disease.

$d0181$ **(including** *alcohol***)** stating that drinking too much alcohol damages various tissues, especially the liver or the heart.

After reading these two documents, the user was not satisfy fully his/her interest since the documents do not mention the causality between the calorie of alcohol and fat directly. If this does not satisfy one's interest, then the user begins to select and read another documents linked from already read documents for getting new information about Q_g. MACLOD helps this complementary reading process with a 2-dimensional interface where a user can piece out the whole relations among documents of obtained plans. That is, user can pick other document, that complements already-read documents, for reaching the satisfaction of her/himself.

The following steps, for example, are as follows. In Figure 5.2, $d0611$ and $d0426$ are linked from $d1459$, and $d1802$ and $d0576$ are linked from $d0181$. Here, because the user wanted to know the limit amount of alcohol to drink, the user was satisfied by reading $d0611$ that states the adequate quantity of alcohol per day. Also, $d0576$ stating the ideal quantity of calorie per day satisfied the user further because his potential interest was in diet. Thus, MACLOD can supply complementary documents step by step according to the user's interests until the user gets satisfied.

5.3 The Answering System Compared with MACLOD

We compared the performance of MACLOD with the following typical search engine for question answering. We call this search engine here a Vector-based FAQ-finder ($VFAQ$ in short hereafter).

The Procedure of VFAQ

Step1') Prepare keyword-vector v_x for each question Q_x in C_{doc}.
Step2') Obtain keyword-vector v_Q for the current query Q_g.
Step3') Find the top N' keyword-vectors prepared in 1'), in the decreasing order of product value $v_x \cdot v_Q$, and return their corresponding answers.

Here, a keyword-vector for a query Q is formed as follows: Each vector has $|Key|$ attributes (Key was introduced in 3.2 as the candidate of keywords in C_{doc}), each taking the value of TFIDF[6] in Q, of the corresponding keyword. Each vector v is normalized to make $|v| = 1$. For example, for query Q_g $\{alcohol,\ warm\}$ (or a question which is put into G: $\{alcohol,\ warm\}$), the vector comes to be $(0, 0.99, 0, \cdots, 0, 0.14, 0, 0, \cdots)$ where 0.99 and 0.14 are the normalized TFIDF values of "alcohol" and "warm" in Q_g. Elements of value 0 here correspond to terms which are in Key but not included in Q_g. Supplying N' documents in Step 3') is for setting the condition similar to MACLOD so that a fair comparison becomes possible.

5.4 Result Statistics

The experiment was executed for 5 subjects from 21 to 30 years old. This means that the subjects were of the near age to the past question askers of *Alice*.

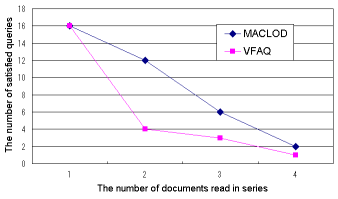

Fig. 2. Statistical results.

A popular method for evaluating the performance of a search engine is to see *recall* (the number of relevant documents retrieved, divided by the number of relevant documents to user's query in C_{doc}) and *precision* (the number of relevant documents retrieved, divided by the number of retrieved documents). However, this traditional manner of evaluation is not appropriate for evaluating MACLOD, because it does not output a sheer list of most relevant documents to the query. In the traditional evaluation, it was regarded as a success if user gets satisfied by reading a few documents which are highly ranked in the output list. On the other hand, MACLOD aims at satisfying a user who reads some documents along the pathways, rather than a few best document. Therefore, this section presents an original way of evaluation for MACLOD.

Here, 42 queries were entered. This seems to be quite a small number for the evaluation data. However, we compromised with this size of data because we aimed at having each subject evaluate the returned answer in a natural manner. That is, in order to have the subject report whether s/he was really satisfied with the output, the subject must enter his/her real ïareinterest. Otherwise, the subject has to imagine an unreal person who asks the rare query and imagine what the unreal person feels with the returned answers. Therefore we restricted to a small number of queries entered from real novel interests.

The overall result was shown in Figure 5.4. The horizontal axis means the number of documents read in series and the vertical axis means the number of satisfied queries. According to the subjects, MACLOD did better than VFAQ, especially for novel queries. For $x = 1$, MACLOD and VFAQ equally satisfied 16 queries. On the other hand, for $x = 2$, MACLOD satisfied 12 queries, whereas VFAQ satisfied 4 queries. And for $x = 3$, MACLOD satisfied 6 queries, whereas VFAQ satisfied 3 queries. Finally, fot $x \geq 4$, MACLOD and VFAQ satisfied 3 queries. Thus, the superiority of MACLOD for x greater than 1 came to be apparent. In all cases, VFAQ obtained redundant documents, i.e., document of similar contexts, equally relevant to the query.

Thsese results can be summarized that novel queries for C_{doc} were anwered satisfactory by MACLOD. Answers in the form of document-combination visualized by MACLOD came to be easy to read and browse along the links according to the subject, and the presented answers were meaningful for the user.

5.5 Comparison with Other Methods

Among the rare systems which combine documents for answering novel query, Hyper Bridges[10] and *NaviPlan*[11] produce a plan of user's reading of documents. They present a plan made of sorted multiple documents, and a user who reads them in the order as sorted by Hyper Bridges or *NaviPlan* incrementally refines one's own knowledge until one learns the meaning of the entered query. A plan made by these tools is a *serial* set of documents, which guides the user to an understanding of query starting from a beginner's knowledge, in the order presented by the system. As a result, neither *NaviPlan* nor Hyper Bridges they can obtain an appropriate document to be read last, i.e., the document to directly reach the goal (i.e. answer the query), in all the examples above where multiple documents are required to be mixed to answer the query. On the other hand, the combination retrieval and its advanced version MACLOD makes a *complementary* set of documents, supplementing the content of each other for making a satisfactory answer as a whole. User may read documents in an obtained document-set in any order as s/he likes. Especially, MACLOD gives user more flexible search interface than the original combination retrieval.

Let us here show the merit of MACLOD compared with the previous combination retrieval. In short, the merit is that user can select documents matching with his/her interest, reactively reflecting the context of documents read already. The fair extension of the combination retrieval to be compared with MACLOD is to have as many document-sets as obtained in MACLOD. In such an output style, it will be difficult to control the context of the documents to read. That is, the order of sets sorted on *cost* does not always correspond to user's interest and often bothers user with compelling to read the document-sets in an undesired order. In this example, if the user feels d1459 mismatching to his/her context, s/he will not reach any satisfactory document-set in the list. Neither a MACLOD-like style output as in Figure5.2 makes things better, in this case because d1459 is shared by all the sets. In all trials for obtaining and showing highly ranked document-sets of the combination retrieval, the user was fixed to the context bound by a c̈entrald̈ocument as d1459 whether desiring or disgusting the situation. From this problem with the combination retrieval, we can point the two-fold merit of MACLOD.

1. Due to discarding documents already appeared many times in the output document-sets in the process (see Section 4), MACLOD can include document-sets of various contexts in the output. This enables the user to choose suitable contexts reactively in the search process.
2. The graphical output makes the context-control easier, because the links between nodes (documents) represent the complemantary relations (i.e., as documents to be read together) between contexts. If user feels a document misleading to him, s/he can open a document linked from the current document without feeing a sudden departure from the current context.

6 Conclusions

The combination retrieval, a method to obtain a set of documents for answering a novel query is fully described and its visual interface MACLOD is introduced. Combination retrieval presents user with a set of, not a single, documents for answering a new query unable to be answered by one past answer to a past query. The MACLOD interface supplies a user with a further comfortability in acquiring novel knowledge. MACLOD allows user to efficiently alter a part of the reading-plan (i.e. document-set) s/he is currently following, improving his/her satisfaction. This effect works especially if the interest is novel i.e., if the context is too particular to be captured by past Q&A's.

References

1. Hadamard, J: *The Psychology of Invention in the Mathematical Field*. Princeton University Press, 1945.
2. Swanson, D.R., Smalheiser, N.R.: An Interactive System for Complementary Literatures: a Stimulus to Scientific Discovery. *Artificial Intelligence*, Vol. 91, 183–203, 1997.
3. Matsumura, N., and Ohsawa, Y.: Combination Retrieval for Creating Knowledge from Sparse Document Collection, *Proc. of Discovery Science*, 320–324, 2000.
4. Brookes, B. C.: The foundations of information science, *Journal of Information Science*, 2, 125–133, 1980.
5. Porter, M.F.: An Algorithm for Suffix stripping. *Automated Library and Information Systems*, Vol.14, No.3, 130–137, 1980.
6. Salton, G. and Buckey, C.: Term-Weighting Approach in Automatic Text Retrieval, *Reading in Information Retrieval*, 323–328, 1998.
7. E. Charniak and S.E. Shimony: Probabilistic Semantics for Cost Based Abduction. *Proc. of AAAI-90*, 106–111, 1990.
8. Ohsawa, Y. and Yachida, M.: An Index Navigator for Understanding and Expressing User's Coherent Interest, *Proc. of IJCAI-97*, 1: 722–729, 1997.
9. Nonaka,I. and Takeuchi, H.: *The Knowledge Creating Company*, Oxford University Press, 1995.
10. Ohsawa, Y., Matsuda, K. and Yachida, M.: Personal and Temporary Hyper Bridges: 2-D Interface for Undefined Topics, *J. Computer Networks and ISDN Systems*, 30: 669–671, 1998.
11. Yamada, S. and Osawa, Y.: Planning to Guide Concept Understanding in the WWW. *AAAI-98 Workshop on AI and Data Integration*, 121–126, 1998.
12. Ohsawa, Y. and Ishizuka, M.: Networked Bubble Propagation: A Polynomial-time Hypothetical Reasoning Method for Computing Near-optimal Solutions, *Artificial Intelligence*, Vol.91, 131–154, 1997.