Available online at www.sciencedirect.com

Knowledge-Based Systems 18 (2005) 327–333

www.elsevier.com/locate/knosys

Combination retrieval for creating knowledge from sparse document-collection

Naohiro Matsumura[a,b,*], Yukio Ohsawa[a,c,1], Mitsuru Ishizuka[b,2]

[a]*PRESTO, Japan Science and Technology Corporation, Kawaguchi Center Building, 4-1-8 Honcho, Kawaguchi-Shi, Saitama 332-0012, Japan*
[b]*Graduate School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan*
[c]*Graduate School of Business Science, University of Tsukuba, Tokyo, Japan*

Received 13 January 2003; accepted 16 March 2005
Available online 31 May 2005

Abstract

With the variety of human life, people are interested in various matters for each one's unique reason, for which a machine maybe a better counselor than a human. This paper proposes to help user create novel knowledge by combining multiple existing documents, even if the document-collection is sparse, i.e. if a query in the domain has no corresponding answer in the collection. This novel knowledge realizes an answer to a user's unique question, which cannot be answered by a single recorded document. In the Combination Retriever implemented here, cost-based abduction is employed for selecting and combining appropriate documents for making a readable and context-reflecting answer. Empirically, Combination Retriever obtained satisfactory answers to user's unique questions.
© 2005 Published by Elsevier B.V.

Keywords: Information retrieval; Cost-based abduction; Knowledge creation

1. Introduction

People are interested in personal and unique matters, e.g. very rare health condition, friction with friends, etc. They often hesitate to consult a human about such unique matters, and worry in their own minds. In such a case, entering such interests to a search engine and reading the output documents is a convenient way which may serve satisfactory information.

However, a document-collection of a search engine, even though they may seem to include a lot of documents, is too sparse for answering a unique question: They have only past information not satisfactory for answering novel queries. For overcoming this situation, a search engine is desired to help user *create* knowledge from sparse documents.

For this purpose, we propose a novel information retrieval method named *combination retrieval*. The basic idea is that an appropriate combination of existing documents may lead to creating novel knowledge, although each one document may be short of answering the novel query. Based on the principle that combining ideas triggers the creation of new ideas [1], we present a system to obtain and present an optimal combination of documents to the user, optimal in that the solution forms a document-set which is the most readable (understandable) and reflecting the user's context.

The remainder of this paper is organized as follows. In Section 2, the meaning of combination retrieval in this paper is shown by comparison with previous information retrieval methods. The mechanism of the implemented system, Combination Retriever, is described, in Section 3. We show the experiments and the results in Section 4, showing the performance of Combination Retriever for medical counseling question-and-answer documents.

* Corresponding author present address: Graduate School of Economics, Osaka University, 1–7 Machikaneyama, Toyonaka, Osaka, 560–0043, Japan
E-mail addresses: matumura@econ.osaka-u.ac.jp (N. Matsumura), ohsawa@q.t.u-tokyo.ac.jp (Y. Ohsawa), ishizuka@miv.t.u-tokyo.ac.jp (M. Ishizuka).
[1] Present address: Department of Creative Informatics, Graduate School of Information Science and Technology, The University of Tokyo
[2] Present address: Department of Quantum Engineering and System Science, School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656

0950-7051/$ - see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.knosys.2005.03.003

2. Previous methods for answering a query

A question–answering 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.' However, for making such an answer, the system should have medical knowledge relevant to user's query and arrange essential parts of the knowledge. It is not realistic to implement knowledge wide enough to be applied to unique user interests.

Another approach for answering a query 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 human knowledge directly. 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 do not satisfy novel interests as in Section 1.

For a point-hitting response to a query, the approach represented by FAQ (Frequently Asked Question) finder [2][3] selects a past human question-and-answer record (*client's question, counselor's answer*) of which the *client's question* is the closest to the current user's question. The matching of the current and a past query is executed between keywords and sentence types of the two questions. Thus, FAQ finder takes advantage of the professional counselor' knowledge, which is presented for advising a client using easy words, but is hardly implemented as a knowledge-base. It has an additional merit that the user can enter the query in natural language.

Although such a memory-based processing of natural language entry of queries is effective as well as memory-based translation [3], unique queries are too novel to match one past query. Thus, answering a novel query remained an open problem.

3. The process of Combination Retriever

For realizing combination retrieval, we need a method for selecting meaningful documents which, as a set, serve a good (readable and meaningful) answer to the user. Here, we show our approach implemented as a system called *Combination Retriever*, where abductive inference is used for the selection of documents to be combined.

[3] The FAQ Finder system: http://faqfinder.ics.uci.edu/

3.1. The outline of the process

The process of Combination Retriever is as follows:
The process of Combination Retriever

Step (1) Accept user's query Q_g.

Step (2) Obtain G, a word-set representing the goal user wants to under-stand, 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) for and the effect (words to be subsequently understood by reading D_x) of reading document D_x.

Step (4) Obtain h, the optimal hypothesis-set which derives G by being 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 by other information source than the document-collection C_{doc}, for reading the documents in S.

Step (5) Show the documents in S to the user.

Before going into the details, let us mention here that the intuitive meaning of the abductive inference is to obtain the conditions for understanding user's goal G. Those conditions include the documents to read (S) for understanding G, and necessary knowledge (K) for reading those documents. That is, S means the document-combination we aim to present to the user.

3.2. The details of Combination Retriever'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 for all the documents in C_{doc}, is obtained and fixed. Here, words are stemmed as in [7] and stop words ('does', 'is', 'a'…) are deleted, and then a constant number of words of the highest TFIDF values [8] (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 Section 3.1.

Steps (1) and (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’ as keywords (some are deleted as stop words).

Step (3) (Make Horn clauses from documents): For the abductive inference in Step (4) of Section 3.1, knowledge-base Σ is formed of *Horn clauses*. A Horn clause is a clause in Eq. (1), which means that y becomes true under the condition that all x_1, $x_2,\ldots,x_n$ are true, where variables x_1, $x_2,\ldots,x_n$ and y are atoms each of which corresponds to an event occurrence. A Horn clause can describe causes (x_1, $x_2,\ldots,x_n$) and their effect (γ) simply

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

In Combination Retriever, 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:

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

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. β_1, $\beta_2,\ldots,\beta_{mx}$ are the *conditional terms* of D_x, which should be understood for reading and understanding D_x. That is, one who knows words β_1, $\beta_2,\ldots,\beta_{mx}$ 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 α, $\alpha_2,\ldots$ are obtained as terms in $G \cap$(*the keywords of* D_x). This means that the effect of D_x is determined by 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 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 Retriever as in Section 4.1.

Then, the keywords of D_x other than the effect terms above form the conditional terms β_1, $\beta_2,\ldots,\beta_{mx}$ Finally, Horn clauses are obtained as

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

meaning that one knowing β_1, $\beta_2,\ldots,\beta_{mx}$ can read D_x and understand all the effect terms α_1, $\alpha_2,\ldots$ by reading D_x.

Step (4) (Cost-based abduction for obtaining the documents to read): We employ *cost-based abduction* (CBA, hereafter) [6], an inference framework for obtaining solution h of the least $|K|$ in Section 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 Eqs. (5) and (6). We deal with Σ composed of causal rules, expressed in Horn clauses mentioned above

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

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

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

Eq. (4) represents the selection of h to be minimal, i.e. of the lowest-cost hypothesis-set $h(\subset H)$, where cost denoted *cost*(h) is the sum of *weights* of hypotheses in h. The weights of hypotheses in H, which axe 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 [10]. In Combination Retriever, 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 (learns) a conditional term in *Key*

In giving weights, 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 h is $S \cup K$. It might be good to give values between 0 and 1 to hypotheses of Type 2, earch 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}. Also, it might seem to be necessary to give weights to hypotheses of Type 1, each value representing the cost of reading each document. However, it is not necessary 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 he/she should read in the document. If we sum the heterogeneous difficulties, both of reading documents and of learning words, the meaning of the solution cost would become rather confusing.

3.3. An example of Combination Retriever’s execution

For example, Combination Retriever 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 here, of D_1, D_2, and D_3 of Type 1 each weighted 0, and 'cirrhosis,' 'cell,' 'disease,' 'maxijuana,' '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) 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 he should limit drinking 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 Retriever, 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 words. 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 create new knowledge [11], to satisfy his/her novel interest.

4. Experimental evaluations

4.1. The experimental conditions

Combination Retriever was applied to C_{doc} of 1320 question–answer pairs from a health care question–answering service on WWW (*Alice*, http://www.alice.columbia.edu). Past clients of *Alice* asked about personal anxiety or interest in health and a medical counselor answered them.

Applying Combination Retriever to such a collection of past questions and answers made by a human counselor and client makes a fair experiment, because the vocabulary gap [12] between the machine counselor and the subject client does not disturb the client's understanding of the answer, thanks to the human counselor's effort to make easy answers. Also, the small number as 1320 documents is a suitable condition for evaluating Combination Retriever for a sparse document-collection which is insufficient for answering novel queries.

When a user enters a query in a word-set or a sentence, Combination Retriever obtains solution h and shows the output S in h as in Fig. 1. In this case, input $\{$*alcohol, cancer, liver*$\}$ was entered as query Q_g.

Combination Retriever is fully implemented in a Pentium Pro 300 MHz machine with 256 MB memory. Although CBA is time-consuming because of its NP-completeness, most answers in the experiments were returned within 10 s from the entry of query by high-speed abduction as in [13]. Queries from users included four or less terms in *Key*, due to which the response time was below 10 s. This quick response comes also from the goal-oriented construction of Horn clauses shown in Section 3.2, and from the absence of inconsistencies among hypotheses which might have slowed the inference.

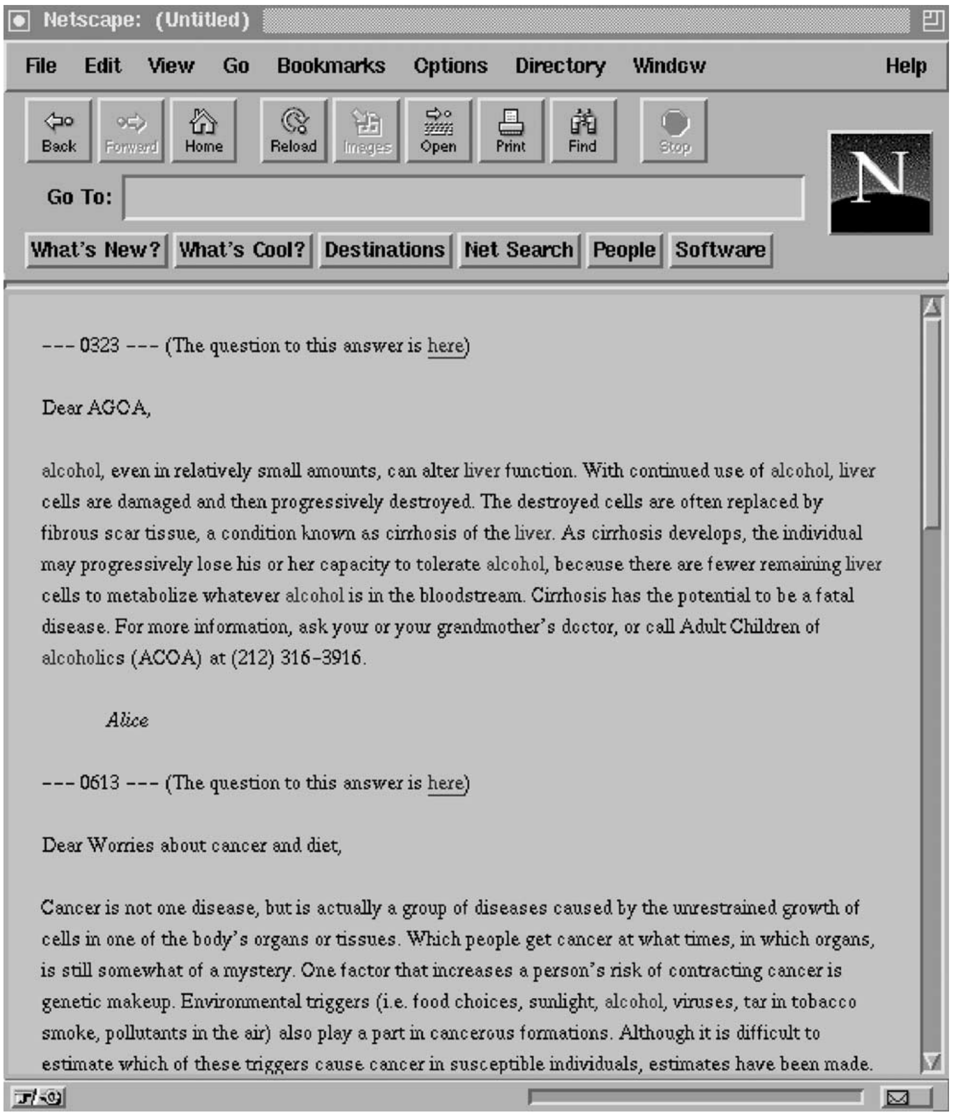
4.2. The answering system compared with Combination Retriever

We compared the performance of Combination Retriever 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

Step ($1'$) Prepare keyword-vector v_x for each question Q_x in C_{doc}.
Step ($2'$) Obtain keyword-vector v_Q for the current query Q_g.
Step ($3'$) Find the top M keyword-vectors prepared in Step ($1'$), in the decreasing order of product value $v_x \cdot v_Q$, and return their corresponding answers (each denoted A_x). Here, M is the number of documents in the answer of Combination Retriever for the same query Q_g.

Here, a keyword-vector for a query Q is formed as follows: Each vector has $|Key|$ attributes (*Key* was introduced in the earlier section as the candidate of keywords in C_{doc}), each taking the value of TFIDF [8] 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,…,0, 0.14, 0, 0,…) where 0.99 and 0.14 are the normalized


Netscape: (Untitled)
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Go To:
What's New? What's Cool? Destinations Net Search People Software

--- 0323 --- (The question to this answer is here)

Dear AGOA,

alcohol, even in relatively small amounts, can alter liver function. With continued use of alcohol, liver cells are damaged and then progressively destroyed. The destroyed cells are often replaced by fibrous scar tissue, a condition known as cirrhosis of the liver. As cirrhosis develops, the individual may progressively lose his or her capacity to tolerate alcohol, because there are fewer remaining liver cells to metabolize whatever alcohol is in the bloodstream. Cirrhosis has the potential to be a fatal disease. For more information, ask your or your grandmother's doctor, or call Adult Children of alcoholics (ACOA) at (212) 316-3916.

Alice

--- 0613 --- (The question to this answer is here)

Dear Worries about cancer and diet,

Cancer is not one disease, but is actually a group of diseases caused by the unrestrained growth of cells in one of the body's organs or tissues. Which people get cancer at what times, in which organs, is still somewhat of a mystery. One factor that increases a person's risk of contracting cancer is genetic makeup. Environmental triggers (i.e. food choices, sunlight, alcohol, viruses, tar in tobacco smoke, pollutants in the air) also play a part in cancerous formations. Although it is difficult to estimate which of these triggers cause cancer in susceptible individuals, estimates have been made.

Fig. 1. An output of Combination Retriever, showing two past answers 0323 and 0613 (document IDs in C_{doc}) for input query (*alcohol, cancer, liver*}.

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 M documents in Step (3′) is for setting the condition similar to Combination Retriever so that a fair comparison becomes possible.

It is possible to make VFAQ more similar to FAQ finder as in [2] by replacing document (question–answer pair) D_x with Q_x, question in D_x, in the procedure above. However, we do not do so for two reasons. First, if we name this version VFAQ0, VFAQ0 is less fair than VFAQ for comparing with Combination Retriever because Combination Retriever chose D_x in stead of Q_x for matching Q_g.

It is also possible to include a giving-up option, i.e. output 'I have no answer for you' if $v_x \cdot v_Q$ is less than a fixed allowable threshold for v_x of all D_x in C_{doc}. However, in order to avoid the risk of unreasonably discounting VFAQ, we do not take this option. That is, M documents of the largest $v_x \cdot v_Q$ in C_{doc}, were always shown and compared with the answer of Combination Retriever.

Comparing with Combination Retriever's minimization of *cost*(h), VFAQ has no strategy for reducing the redundancy, i.e. for reducing output documents of similar contents.

4.3. Result examples

Let us show some examples, which show why and how Combination Retriever serves merits to user.

Query 1: A subject worrying about his fat, but who likes to drink alcohol, entered query Q: {*fat, alcohol, calorie*} instead of a question 'Does the calorie of alcohol produce fat in my body?'

Answer 1: Documents D_4 plus D_5 were obtained by Combination Retriever, and the user came to understand that alcohol gains his fat anyway.

D_4 (including *alcohol* and *calorie*) saying that alcohol contains much calorie, but does not change into fat in human tissues.

D_5 (including *alcohol* and *fat*) saying that alcohol disturbs fat in human tissues from being metabolized.

On the other hand, VFAQ obtained D_4 and another document of similar content to D_4, because 'calorie' had much greater TFIDF value than 'fat' in D_5. This example shows a typical advantage of Combination Retriever because D_5 disallows the user to drink as he likes, considering significant effects not considered in D_4 which looks like allowing the user to drink.

Query 2: The user entered {*alcohol, warm*} for knowing if drinking alcohol makes one warm if it is cold.

Answer 2: Combination Retriever returned D_6 and D_7 shown below. 'metabolize' and 'fat' shared by D_6 and D_7 reduced the total cost of the combined answer here. The user read these, and found that metabolizing fat is the essential point for his question. We can regard this point as the discovered context as in the example of Section 3.3. As a result, the user understood that he/she should not drink alcohol for keeping himself warm, but rather eat fat-rich food and do exercise.

D_6 (including *warm*), saying that an effective way for keeping oneself warm is eating fat and exercising. Fat is easily metabolized and produces fever.

D_7 (including *alcohol*) saying that alcohol disturbs fat in human tissues from being metabolized (the same document as D_5)

On the other hand, VFAQ obtained D_6 and a document recommending exercise, similarly to D_6.

Query 3: 'Is there any drug to reduce the risk of cancer ?'

Answer 3: G came to be {*risk, cancer, drug*}. The following documents D_8 and D_9 were returned by Combination Retriever. Here, *aspirin* was shared as a conditional term in D_8 and D_9 although not in Q_g, and reduced the total cost of the combined answer. The subject understood that lifestyle is more important than drug for reducing cancer risks.

D_8 (including *risk* and *cancer*), saying that aspirin reduces the risk of cancer.

D_9 (including *drug* and *cancer*) saying that aspirin is an effective drug for many diseases including cancer, but may irritate the stomach and bleed intestinal ulcers. A healthy lifestyle is more important than drugs.

On the other hand, VFAQ returned redundant two documents, both stating about general risks of cancer.

These three examples show that Combination Retriever obtains documents, which reflects the discovered context underlying user's query, and form a context-matching and easy-to-read answer as a whole. Note that the subject user's queries here were unique for C_{doc}, i.e. not similar to any past question.

4.4. Other methods

Among the rare systems which combine documents, Hyper Bridges [4] and *NaviPlan* [5] 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 word or phrase.

A plan made by these tools is a *serial* set of documents, which guides the user to an understanding of query words starting from a beginner's knowledge, if the user reads the documents in the order presented by the system. As a result, either *NaviPlan* or Hyper Bridges obtains no result, because they cannot obtain the 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, Combination Retriever makes a *parallel* set of documents, supplementing the content of each other for making a satisfactory answer as a whole. User may read them in any order as he/she likes.

4.5. Result statistics

The test was executed for five subjects from 21 to 30 years old accustomed to using a Web browser. This means that the subjects were of the near age to the past question askers of *Alice* and entered queries into the CGI interface and read the answers of Combination Retriever smoothly.

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 Combination Retriever, because the retrieved document-set is not a 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, Combination Retriever aims at satisfying a user who reads the combination of all the output documents, rather than a few best document. Therefore, this section presents an original way of evaluation for Combination Retriever.

Here, 63 queries were entered. This seems to be quite a small number for the evaluation data. However, we compromised with this size of data for two reasons. First, we aimed at having each subject evaluate the returned answer in a natural manner. That is, in order to have the subject report whether he/she was really satisfied with the

Table 1
Result statistics

	$M=1$	$M=2$	$M\geq 3$	Total
Combination Retriever	16/22	21/29	6/12	423/63
VFAQ	13/22	13/29	0/12	26/63

output of Combination Retriever, the subject must enter his/her real anxiety or interest. Otherwise, the subject has to imagine an unreal person who asks the query and imagine what the unreal person feels with the returned answers. Therefore, we restricted to a small number of queries entered from real interests. Second, the results show significant superiority of Combination Retriever as follows, even though the test data was small.

The overall result was that Combination Retriever satisfied 43 of the 63 queries, while VFAQ satisfied only 26 queries. Next, let us show more detailed results, which show that Combination Retriever works especially for novel queries.

The dependence of performance on M: According to the subjects, Combination Retriever did better than VFAQ, especially for important queries, important in that they consulted a machine instead of a human counselor due to the *uniqueness* of the query. In such a case, Combination Retriever returned a set of multiple documents because any single past document could not answer the query. Let us show this, on the performance dependence on M, the number of required output documents of Combination Retriever as defined in Section 4.2. M implies the uniqueness of the query, i.e. how it is difficult to answer the query with an existing document.

The result statistics is shown in Table 1. For the 22 queries of $M=1$, Combination Retriever satisfied 16 queries, whereas VFAQ satisfied 13 queries. On the other hand, for the 29 queries of $M=2$, Combination Retriever satisfied 21 queries, whereas VFAQ satisfied 13 queries. Finally, for the 12 queries of $M\geq 3$, Combination Retriever satisfied six queries, whereas VFAQ satisfied not a query. Thus, the superiority of Combination Retriever for the larger M came to be the more apparent. In all cases, VFAQ obtained redundant documents as in the examples presented above.

These results can be summarized as that unique queries for C_{doc} were answered satisfactorily by Combination Retriever. Answers in the form of document-combination by Combination Retriever came to be easy to read according to the subjects, and the presented answers were meaningful for the user.

5. Conclusions

We proposed to help user create novel knowledge, by combining and presenting multiple existing documents. This novel knowledge realizes an answer to user's unique question, which cannot be answered by a single document.

This high-performance comes from obtaining minimal-cost hypothesis in CBA. That is, a document-set in a meaningful context can be obtained, because CBA discovers relevant context according to user's query, by minimizing the number of conditional terms for reading output documents. This means that the user and the system can ask and answer under a meaningful context, which supports a meaningful communication. From such a novel and meaningful context presented, the user can create new knowledge which realizes a satisfaction of his/her unique interest. This is a significant by-product of minimizing the cost of output-documents for obtaining an answer easy to read.

References

[1] J. Hadamard, The Psychology of Invention in the Mathematical Field, Princeton University Press, Princeton, New Jersey 08544 USA, 1945.
[2] R. Burke, K. Hammond et al., Question answering from frequently-asked question files: experiences with the FAQ finder system, Department of Computer Science Technical Report TR-97-05, University of Chicago (available from http://infolab.cs.uchicago.edu/faqfinder/), 1997.
[3] S. Sato, M. Nagao, Toward memory-based translation, Proceedings of COLING-90 3 (1990) 247–252.
[4] Y. Ohsawa, K. Matsuda, M. Yachida, Personal and temporary hyper bridges: 2-D interface for undefined topics, Journal of Computer Networks and ISDN Systems 30 (1998) 669–671.
[5] S. Yamada, Y. Ohsawa, Planning to guide concept understanding in the WWW, in: Workshop Notes on AAAI-98 Workshop on AI and Data Integration (1998).
[6] E. Charniak, S.E. Shimony, Probabilistic semantics for cost based abduction, Proceedings of AAAI-90 (1990) 106–111.
[7] M.F. Porter, An algorithm for suffix stripping, Automated Library and Information Systems 14 (3) (1980) 130–137.
[8] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing and Management 14 (1988) 513–523.
[10] Y. Ohsawa, M. Yachida, An index navigator for understanding and expressing user's coherent interest, Proceedings of IJCAI-97 1 (1997) 722–729.
[11] I. Nonaka, H. Takeuchi, The Knowledge Creating Company, Oxford University Press, Oxford OX2 6DP, England 1995.
[12] G.W. Furnas, T.K. Landauer, L.M. Gomez, S.T. Dumais, The vocabulary problem in human-system communication, Communications of the ACM 30 (11) (1987) 964–971.
[13] Y. Ohsawa, M. Ishizuka, Networked bubble propagation: a polynomial-time hypothetical reasoning method for computing near-optimal solutions, Artificial Intelligence (Elsevier) 91 (1997) 131–154.