2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Construction of Local Document Management System based on Associative Search

Yoshimasa Kasagi [*1], Toru Yamaguchi[*1*2], Yasufumi Takama[*1*2]

[*1]Tokyo Metropolitan Institute of Technology
[*2]Japan Science and Technology Corporation (JST)
Address: 6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan
Email: Kasagi@krectmt3.tmit.ac.jp

Abstract - As the information that can collect from the web to local database is increasing, we propose a system that can suggest related local documents when new document arrives. We also propose for constructing an association dictionary using web search engines for similarity calculation. The prototype system is also developed, which is described in detail.

1. Introduction

Although it becomes easy to obtain documents from the web and to store in a local database, we often forget the stored documents when we actually need them. In such a case, we never try to search them. In order to solve this problem, we propose the system with which documents are passively searched when we get new information from the web, based on associative dictionary. By the passive-style search that is triggered without user's explicit information need, system can increase the opportunities of showing documents to a user. Therefore, the system is expected to help users get new idea and knowledge. Conventional local document search systems have been insufficient for retrieving related documents, because only files that contain an input word are displayed. In this paper, we propose for retrieving local documents related to incoming document based on the association dictionary that is created by using web search engine. The prototype system is also developed, which is described in detail.

2.Concept of local document management system

A local document management system that we propose consists of local document database, retrieval engine, and associative search dictionary. A local document database stores the documents, in which a user is interested. A retrieve engine calculates the similarity between the new document and documents in local document database, and returns ranked document list.

When designing the document retrieval systems, it is often said that there is trade-off between precision and recall. Compared with the web, in which most of documents are irrelevant to a user's interest, local document database contains only the documents of interest. Therefore, recall is more important than precision for retrieving local documents. From this viewpoint, we use associative search dictionary, because a system can retrieve more local documents than using Boolean match. In this study, we use web search engine to create associative search dictionary. By using web search engine to create associative search dictionary, system can cope with new information and calculate relevance between words quantitatively.

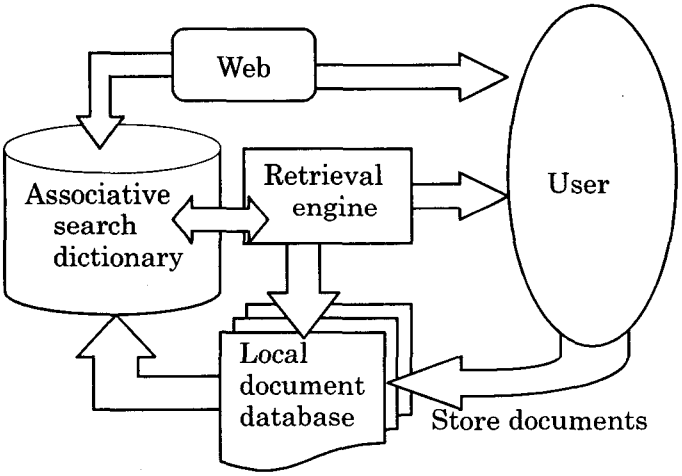

Fig.1 Image of the system

Fig.1 is the image of the local document management system. A user can get various information from the web and store useful information to local as he or she likes. If a user receives information from the web, the system automatically searches related local documents, and suggests them to the user. A user can passively get only the related documents that have been judged as useful and been stored by the user oneself.

3. Associative search dictionary

Word-relevance between word W_A and word W_B, $r(W_A, W_B)$, is defined as Eq.(1).

$$r(W_A, W_B) = \frac{2 \times fc}{fa + fb} \quad (1)$$

Here, fa and fb represent the number of search results of a search engine when each word W_A and W_B is submitted, and fc represents that when both words are submitted together. In this study, we use the web search engine Google (http://www.google.com/).

Based on the relevance between words calculated by Eq. (1), we calculate Relevance between words and documents in order to search the documents related with the words.

As we handle the documents written in Japanese, a Japanese morphological analysis tool 'Chasen' is used to extract nouns from documents (http://chasen.aist-nara.ac.jp/).

Extracted noun group represents the original document.

Table.1 Example of stop word list

Abstract word	Impression, Results, Means, Standard, Months.
Wide meaning word	Purpose, Scale, Materials, Company, Object, Culture,

A stopword list made by manual operation beforehand is also used in order to do an efficient associative search, as there exist many words that are irrelevant as index terms. For example, abstract words or wide meaning words, such as shown in Table1, are selected as stop words.

Relevance between Word W_A and document D : $R(W_A, D)$, is defined as Eq. (2).

$$R(W_A, D) = \frac{1}{M} \sum_{W_D \in D} r(W_A, W_D) \quad (2)$$

Here W_D is an index word extracted from document D, and M is the number of nouns extracted by document D. Relevance between word and document are calculated by Eq. (2), and those data are prepared as database in advance.

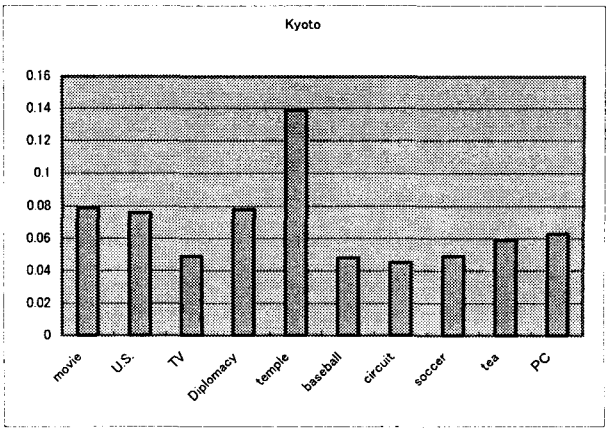

Fig.2 Relevance with "Kyoto"

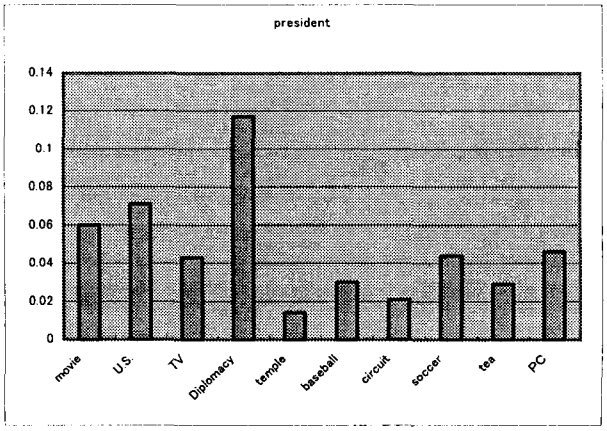

Fig.3 Relevance with "president"

Fig.2 and Fig.3 represent relevance between 10 documents and words "Kyoto" and "president", which are calculated by Eq. (2). Each document contains difference topics like TV, and sports. Fig.2 shows "Kyoto" has higher relevance value with a document about temple than others, and Fig.3 shows "president" has higher relevance values with a document about diplomacy than others. In this way, we can see calculating the relevance using search results of a web search engine is effective. We construct local document management system based on this method.

4. Local document management system

We construct the prototype system that can search local documents based on associative dictionary as described in section2. Fig4 is the main interface window of the local document search system. This system is available on Linux.

When a user pastes new document in text area, and executes "save documents", local documents related with new one are retrieved and ranked according to their similarities in local document area. Relevance between documents is calculated by associative search dictionary as described in section2. By executing "dictionary registration", the index words that are included in new document are added to associative search dictionary, and the modifications become valid since the next search.

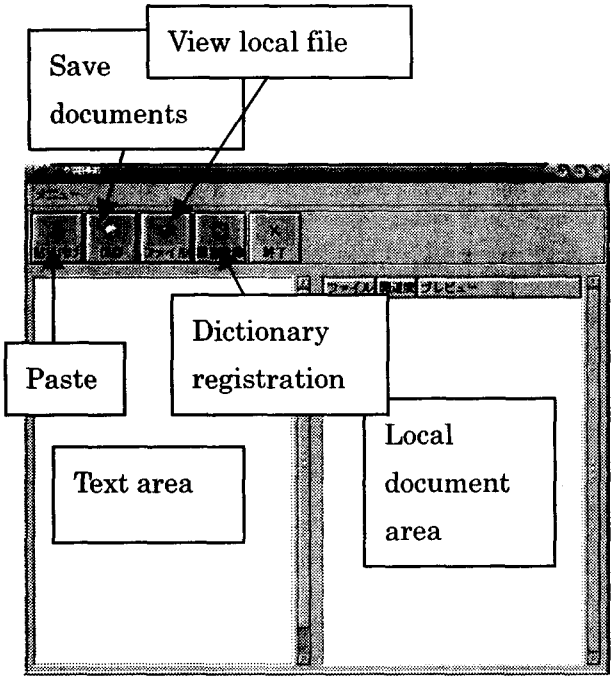

Fig.4.Interface window

5. Execution example

We test the system with associative search dictionary and 50 sample local documents in section 4. When a user saves the new document, extracting new nouns from the document, registering new words, and gathering the relevance data from the web take too much time. It is unrealistic to perform such tasks online, so the system extracts only the nouns that are registered on associative search dictionary in advance.

In the example there are about 350 words in that dictionary. These words are extracted from 50 sample documents prepared beforehand. The contents of 50 sample documents are news articles of sports and government, and the documents about TMIT (Tokyo Metropolitan Institute of Technology).

In Fig.5 we paste the document about digital camera in text area. When we execute save documents, the system extract nouns, retrieve documents based on associative search dictionary, and local documents sorted according to similarity are appeared in local document area. The words marked with a circle in text area are the word stored in associative search dictionary.

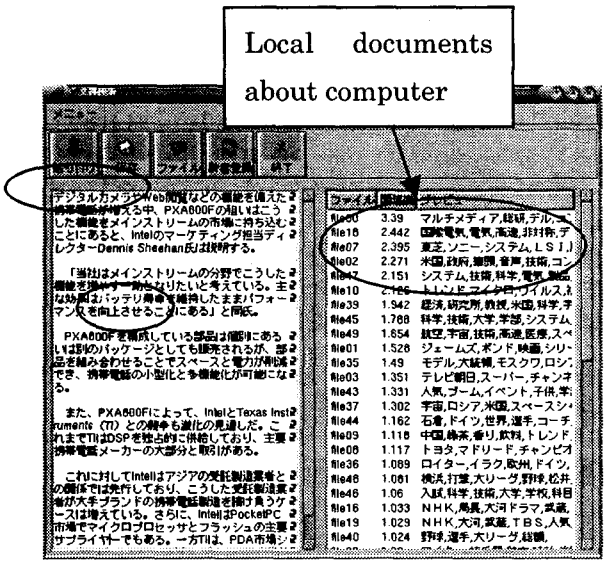

Fig.5 Searching local documents
(Shown with Japanese language)

From the search result, we can see the document about digital camera relates with documents about computer.

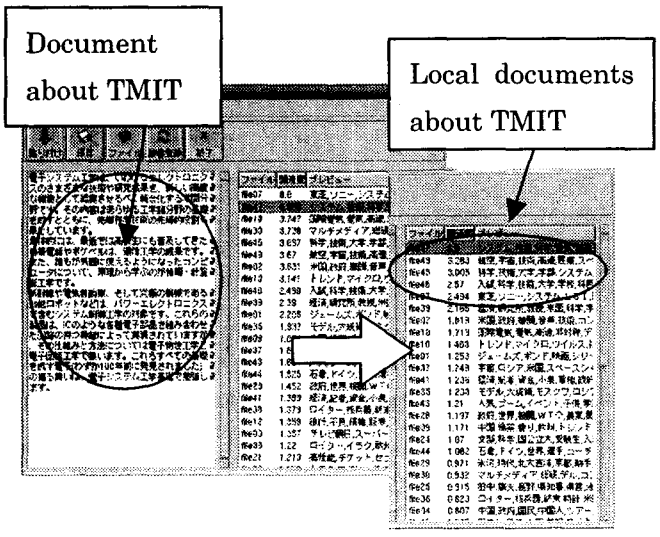

Fig.6 Sorting local documents

This system can also retrieve local documents with a local document. In Fig.6, a local document ranked second in local document area is a document about TMIT. When the document is pasted on the text area, the documents about TMIT are retrieved as relevant. In this way, this system can also search local documents by using a local document as a query.

6.CGI local document management system

In Section 3,we describe the prototype system that is available on Linux. Now we are constructing the CGI system that can use with the usual browser. As noted before, we aim to develop the local document management system, which passively searches local documents, and suggests related documents. CGI system can use anybody easily from various platforms, and cooperate with the web to get new information or to create associative search dictionary by using web search engines.

Fig.7 is the window of CGI system. A user can save the information carelessly without thinking about genre. And this system will suggest related documents to a user according to an incoming document.

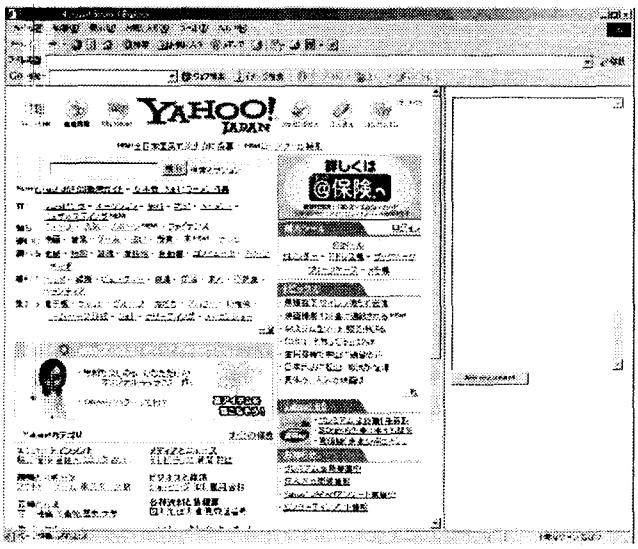

Fig.7 CGI system window

7. Conclusion

Prototype local document management system is implemented, and demonstrated with around 50 sample documents. One of the essential parts of the system is a similarity calculation between documents based on associative search dictionary. It is observed that documents that contain many Net-related words or wide meaning words come to rank higher when there were no documents related with input document. This is because we use the search results of a web search engine to calculate the relevance between words. Improving stopword list will decrease this problem. As another solution, we are now considering the improvement of the index word extraction method so that the words or phrases of specific meanings can be extracted from a document without using a morphological analysis tool. By employing this method the associative search dictionary created by web search engine will be more effective. Furthermore, we are now considering category of the index word. For example, information about time, which is not considered in present method, will be important for retrieving local documents that relate with a certain event.

References

[1] Hirata, Murakami, Nishida : Support for Community Knowledge Sharing with Associative representation and talking-Virtualized-Egos Metaphor, JSAI Journal,vol.15 No,6,pp.1117-1124,2000.

[2] Sunayama, Ohsawa, Yachida : A Search interface with Supplying Search Keywords by Using Structure of User Interest, JSAI Journal, Vol.16 No.2 pp.225-233,2001.

[3] R.Mandala, T.Tokunaga, H.Tanaka, A.Okumura, andK.Satoh : Ad Hoc Retrieval Experiments Using Word Net and Automatically Constructed, NEC, Tokyo Institute of Technology ,TREC7 No48.