Chinese Textual Retrieval Based on Fuzzy Concept Network

Tyne Liang and Ching-chyuan Chang
Department of Computer and Information Science
National Chiao Tung University, Taiwan 30050
E-mail: tliang@cis.nctu.edu.t
Tel: 886-3-5712121 ext. 56632, Fax. 886-3-5721490

Abstract

In this paper, a Chinese document retrieval system based on a fuzzy concept network is proposed and implemented. The system contains two main parts, namely the network module and the query module. The proposed network is realized with concept matrices so that searching through the network is simplified to be a matrix computation. Meanwhile, the relation values among network nodes are computed by proposed weighting functions which take keyword occurrence frequencies and location information into account. On the other hand, the query module is incorporated with a proposed concept -based relevant document retrieval module in order to facilitate relevance feedback. Finally the system is tested with a collection of real Chinese technical documents and qu eries. The experimental results show that the proposed concept-based retrieval yield higher retrieval accuracy rate than the traditional simple keyword-matching retrieval.

1. Introduction

Nowadays many commercial retrieval systems are based on keyword-matching and Boolean logic due to their simple implementation [9]. However, a single word ma contain different meanings within different contexts and one meaning can be described by different words. Besides, many Boolean-logic-based systems cannot efficiently handle those fuzzy queries existed in a real world [7]. To overcome such problem, Lucarella, etc. [6] proposed a fuzzy concept network in which the graded generalization relation among concept nodes is employed and a user could issue a concept instead o f an explicit query term during retrieval process. Later Itzkovich [4] added the additional similarity relation in the network so that concepts can be represented in a more flexible way. However searching among multiple potential paths in a concept network will become complicated when number of concepts increase. Hence a concept matrix proposed b Chen [1] is used to implement a fuzzy concept network. Such matrix not only efficiently solves the implicit relation value existed among concepts but also speeds up document search.

In this paper, a fuzzy concept network retrieval system is proposed and implemented particularly for Chinese textual information. The characteristics of the proposed system are that the relation value computation among network nodes is automatic and a proposed concept-based relevant document module is incorporated with its query module to facilitate relevance feedback. Experimental results show that the proposed module yields higher retrieval accurac rate than a general keyword-based module with respect to different queries and output requirements.

In the following sections, Section 2 will describe an overall structure of the proposed textual retrieval system including the design of keyword -to-concept weighting function, the construct ion of concept hierarchy, the computation of relation values among network links and the implementation of concept matrix. Section 3 describes the proposed concept -based relevant document retrieval module and it will be compared with a keyword -based module. Section 4 is final conclusion.

2. The Fuzzy Concept Network Retrieval System

An overall flowchart of the proposed fuzzy concept-based retrieval system is shown in f igure 1 in which shadowed line depicts document processing, dash line depicts practical on-line query processing, and dotted line depicts relevance feedback processing. In document processing, each document is going to be formatted and stored as formatted textual data in a document database. Then a keyword extraction procedure is applied to extract all keywords of a document so that a document -concept relation value can be computed and stored in the concept-document matrix for subsequent retrieval.

During query processing, an inquiry is expressed as a concept (a broad term generally like a class name which describe a class of objects containing the same attributes), rather than a single particular term. Then the proposed concept matrix computation module i s used to find all relevant documents. If a user is not satisfied with the retrieved documents, he or she may resubmit a new query or choose one interesting document out of the previously

retrieved documents as the input to the concept -based relevant document retrieval module and those documents similar to the input document will be retrieved.

2.1 Keyword-concept weighting functions

During network construction, each concept node of the network is represented with a set of keywords which are extracted from the experimental documents and which are weighted by the keyword -concept weighting functions. Since the documents we used in our experiments are technical reports, so those author-given keywords as well as their occurrence frequencies are collected during keyword extraction. They will be used in a way that the keywords of documents in a concept will be grouped together and weighted by the proposed keyword -concept function. This function (i.e. Equation (2 -1)) takes into account the distribution, the concentration of a keyword as well as its appearance in each individual document and length. $w\_concept\_level_2\ (k_v, c_i)$ for keyword k_v in concept c_i at the second level is calculated to be

$$w\_concept\_level_2(k_v, c_i) = \frac{\sum_{j=1}^{t} LW\_doc(k_v, d_j)}{\frac{T}{2^{Entropy(k_v)}}} \quad (2\text{-}1)$$

where $LW\_doc(k_v, dj)$ is the weight of keyword k_v in document j, t is the number of documents in a concept c_i, T is the total number of keywords appearing in c_i and Entropy(k_v) is defined in Equation (2-1b). $LW\_doc(k_v, dj)$ is defined as:

$$LW\_doc(k_v, d_j) = w_1 \times \frac{df_v}{Sum\_df} + w_2 \times \frac{tf_v}{Sum\_tf} + w_3 \times \frac{keyf_v}{Sum\_keyf} + w_4 \times \frac{length_v}{\max\_length} \quad (2\text{-}1a)$$

where w_1, w_2, w_3, w_4 are input parameters (are constant number 1, 2, 3, 4 respectively in our experiments) and df_v is the number of documents containing keyword k_v in a concept at the second level, tf_v is the number of document titles containing keyword k_v, $keyf_v$ is occurrence frequenc of keyword k_v in keyword sets, $length_v$ is the keyword k_v's length (in terms of the number of characters constituting a keyword); Sum_df, Sum_tf, Sum_keyf are the corresponding summations for all keywords in a document d_j respectively and max_length is the maximal length of all collected keywords.

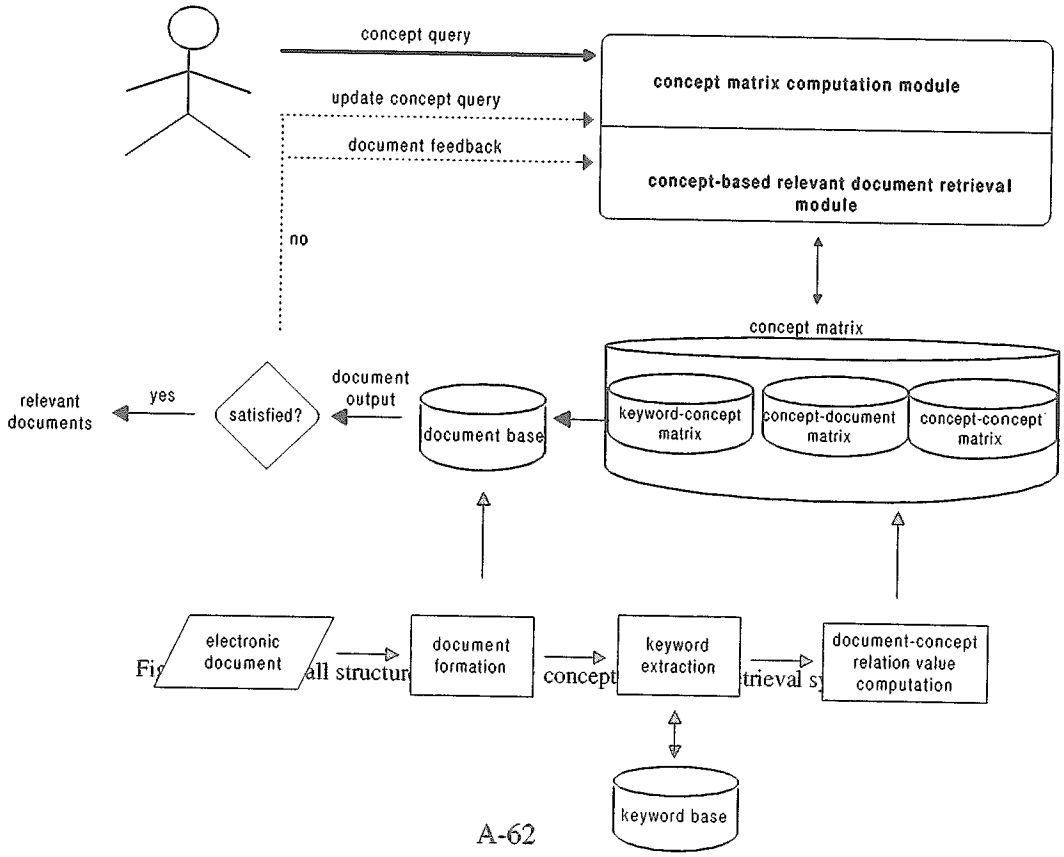
In Equation (2-1) $Entropy\ (k_v)$ is computed as

$$Entropy(k_v) = \sum_{i=1}^{n} \left[\frac{P(k_v, c_i)}{\sum_{i=1}^{n} P(k_v, c_i)} \times \log \frac{\sum_{i=1}^{n} P(k_v, c_i)}{P(k_v, c_i)} \right] \quad (2\text{-}1b)$$

where $P(k_v, c_i)$ indicates the number of documents containing keyword k_v in the concept c_i and n is the number of concepts at second level. Equation (2 -1b) considers the concentration of keywords and it indicates the distribution of keyword k_v in all concepts at the second level. Hence higher concentration will result in lower $Entropy(k_v)$ and consequently higher $w\_concept\_level_2\ (k_v, c_i)$.

2.2 Concept network construction

The proposed network is designed to be a three -level concept hierarchy (refer Figure 2). The nodes at the third (the bottom) level are the do cument nodes which are named by document identifiers. The nodes connected with document nodes are concept nodes at the second level and they are ACM categories in computer science (the categor names are first translated into Chinese terms by the system



designer). The nodes at the first (the top) level are concept nodes which are generated by clustering the concepts at the second level and the clustering is based on fuzzy clustering theory. During generation, the similarity among concepts are calculated with distance measurement [8] and it is defined as Equation (2-2):

$$Sim(c_x, c_y) = \frac{\sum_{i=1}^{u}\left(1 - \frac{|a_i - b_i|}{|a_i + b_i|}\right)}{u} \quad (2\text{-}2)$$

where c_x, c_y are the concepts at the second level, a_i and b_i are the weights of the keyword in concept c_x and concept c_y respectively, u is the number of common keywords in concepts c_x and c_y. It is noticed that the closer the two concepts become, the higher their similarity will be.

As each similarity value between two concepts at the second level is computed, it will be stored in a concept - concept similarity matrix which is defined as following

Definition 2.0: Z is the similarity matrix for constructing the concepts at the first level

$$Z = \begin{bmatrix} Sim(c_1,c_1) & Sim(c_1,c_2) & \cdots & Sim(c_1,c_n) \\ Sim(c_2,c_1) & Sim(c_2,c_2) & \cdots & Sim(c_2,c_n) \\ \vdots & \vdots & \vdots & \vdots \\ Sim(c_n,c_1) & Sim(c_n,c_2) & \cdots & Sim(c_n,c_n) \end{bmatrix}$$

where $Sim(c_x, c_y)$ is the similarity value computed b Equation (2-2) and n is number of concepts.

Then the transitive closure of the matrix is computed in order to obtain the concepts at the first level by using fuzz equivalence relation clustering [5]. A given threshold value is used to decide the number of concepts at the first level.

Once the concepts at the first level are constructed, their keywords are collected by unionizing those keywords in their corresponding concepts at the second level and the keyword weight $w\_concept\_level_1(k_v, c_j)$ in concept c_j at first level is computed by the following equation (2-3):

$$w\_concept\_level_1(k_v, c_j) = \frac{\sum_{i=1}^{s} w\_concept\_level_2(k_v, c_i)}{S} \quad (2\text{-}3)$$

where s is the number of concepts at the second level for c_j and S is the total number of keywords in concept c_j.

Equation (2 -3) concerns mainly the distribution of keywords of the concepts at the second level. That is to say, if the keywords of the conc epts at the first level occur frequently in their corresponding concepts at the second level, then $w\_concept\_level_1(k_v, c_j)$ increases.

Figure 2 shows the result of the proposed three -level concept network in which there are eighteen predefined concepts at the first level and four concepts at the first level. The concepts at the first level are identified with broader Chinese terms by the system designer. The number of concept nodes at the first level in our experiments is generated at threshold value equal to 0.64. This is because the sizes of concept clusters are close to each other at that level.

2.3 Relation Value Computation

There are two types of relations among network links to be computed. One is the generalization value for the nodes between parent and child levels, the other is the similarit value only for the links between two concept nodes at the sibling level.

In the proposed network the similarity values can be computed according to the equation (2 -2) and the generalization value $gw\_level_1\text{-}level_2(c_j, c_i)$ between the first-level concept c_j and the second-level concept c_i is computed by equation (2-4):

$$gw\_level_1\_level_2(c_j, c_i) = \frac{\sum_{v=1}^{m} w\_concept\_level_1(k_v, c_j)}{M} \quad (2\text{-}4)$$

where m is the number of keywords in concept c_i, and M is the total numb er of keywords in the first -level concepts. On the other hand, the generalization values between the second-level concept c_i and the third-level document d_j can be computed by equation (2-5):

$$gw\_level_2\_level_3(c_i, d_j) = \frac{\sum_{v=1}^{r} w\_concept\_level_2(k_v, c_i)}{R} \quad (2\text{-}5)$$

where r is the number of keywords in document d_j and R is the total number of keywords in concept c_i.

2.4 Concept Matrix Construction

In order to speed up search, the proposed network is implemented with concept matrix [1]. There are three matrixes to be built, namely, keyword - concept, concept-concept and concept- document matrixes.

A keyword-concept matrix U (definition 2.1), is used to store keyword weights of concepts at the first level and the second level.

Definition 2.1 : U is a keyword-concept matrix

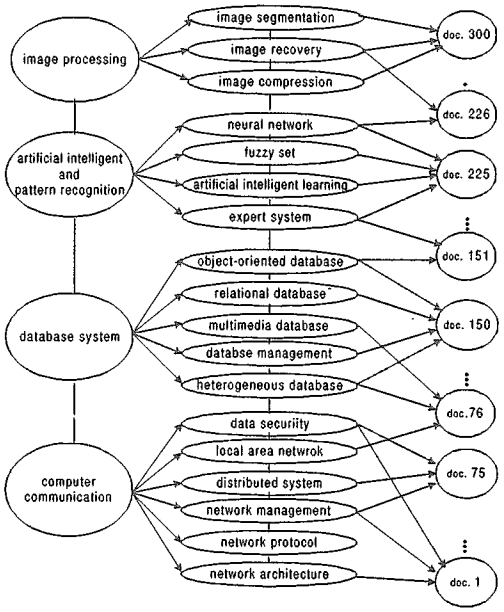


Figure 2: The experimental three-level concept network.

$$U = \begin{bmatrix} w\_concept(k_1, c_1) & w\_concept(k_1, c_2) & \cdots & w\_concept(k_1, c_n) \\ w\_concept(k_2, c_1) & w\_concept(k_2, c_2) & \cdots & w\_concept(k_2, c_n) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ w\_concept(k_m, c_1) & w\_concept(k_m, c_2) & \cdots & w\_concept(k_m, c_n) \end{bmatrix}$$

where $w\_concept(k_i, c_j)$ is the weight of keyword k_i in concept c_j and is computed according to Equations (2.3) and (2.4); n is the number of all concepts; m is the total number of keywords with $1 \le i \le m, 1 \le j \le n$.

The concept-concept matrix T (definition 2.2) is used to store the generalization values between concept nodes. As indicated in [1], the implicative relation values among concepts can be implemented by the computation of the transitive closure T^t of the matrix T.

Definition 2.2: T is a concept-concept matrix

$$T = \begin{bmatrix} gwcc_{11} & gwcc_{12} & \cdots & gwcc_{1n} \\ gwcc_{21} & gwcc_{22} & \cdots & gwcc_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ gwcc_{n1} & gwcc_{n2} & \cdots & gwcc_{nn} \end{bmatrix}$$

where $gwcc_{ij}$ is gw-level$_1$-level$_2$(c_j c_i,) between concepts c_i and c_j and n is the number of all concepts with $1 \le i, j \le n$.

The third matrix is concept-document matrix P (definition 2.3) which is used to store the generalization relation between concepts and documents. Similar to T^t, P^* contains all implicative relations between c oncepts and documents by doing matrix computation, namely $P^* = T^t \otimes P$ [1].

Definition 2.3: P is a concept-document matrix

$$P = \begin{bmatrix} gwcd_{11} & gwcd_{12} & \cdots & gwcd_{1n} \\ gwcd_{21} & gwcd_{22} & \cdots & gwcd_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ gwcd_{m1} & gwcd_{m2} & \cdots & gwcd_{mn} \end{bmatrix}$$

where $gwcd_{ij}$ is $gw\text{-}level_2\text{-}level_3(c_i, d_j)$ (computed by equation (2-5)), m is the number of concepts and n is the number of all documents with $1 \le i \le m, 1 \le j \le n$.

3. Concept-based relevant document retrieval

3.1 Concept-based relevant document retrieval module

As shown in figure 4, a concept-based relevant document module is incorporated to the retrieval system to support relevance feedback. Since a document can be associated with certain number of concepts, relevant documents will be retrieved in the order of their similarity degree between documents and concepts. The similarity is computed b the following equation (3-1):

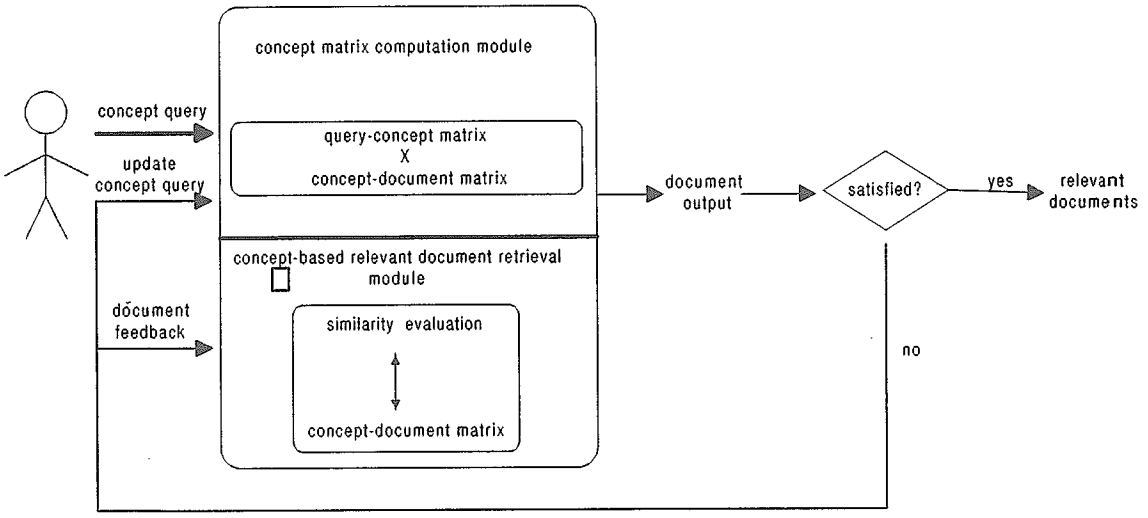


Figure 3: The overview of the concept-based

$$Sim(d_i,d_j)=\begin{cases} 0 & if \quad \sum_{k=1}^{n}(w_{ki}\times w_{kj})=0, \\ \dfrac{\sum_{k=1}^{n}\left(1-\left|w_{ki}-w_{kj}\right|\right)}{n} & if \quad \sum_{k=1}^{n} w_{kj}\neq 0, \end{cases} \qquad (3\text{-}1)$$

where w_{ki} is the weight of document d_i in concept C_k, w_{kj} is the weight of document d_j in concept C_k, both w_{ki} and w_{kj} are computed by Equation (2 -5), n is the number of concepts.

When a retrieved document d_i is chosen by the end-user as the input to the concept-based relevant document retrieval module, then the module computes the similarity (b equation 3-1) between d_i and every other document in concept-document matrix P^*, and retrieve those documents similar to d_i. It is noticed that those retrieved documents are usually associated with the same concept as the input document is, thus relevance of retrieved information will be improved with the proposed module.

3.2 Experiments

The experiments are conducted on UNIX in C language. The network contains total twent -two concepts, four of them are at the top level and eighteen concepts at the second level. For each concept there are twelve to twenty-two technical reports collected as our testing data.

In the test the generalization and similarity relation value of the network are set to be 0.7 a nd the number of output documents to be 20. System expert will judge whether a retrieved document is relevant or not to an inquiry. System performance evaluation is measured in terms of accurac which is defined as:

$$\text{accuracy} = \frac{\text{no. of relevant and retrieved documents}}{\text{no. of retrieved documents}} \qquad (3\text{-}2)$$

To compare with the proposed concept -based retrieval, a general keyword -based retrieval [10] is implemented in such a way that a document-to-document similarity matrix is used t o find the most similar document. A keyword-document matrix F is constructed to store the keyword weights for each document. During relevance feedback, the following similarity function Equation (3 -3) is used to evaluate the similarity between the input document and all the other documents in database. In the equation, f_{is} is the weight of keyword k_s in document d_i and f_{js} is the weight of keyword k_s in document d_j and n is the number of keywords. f_{is} and f_{js} are computed by equation (2-1a).

$$Sim\_keyword(d_i,d_j)=\begin{cases} 0 & if \quad \sum_{s=1}^{n}(f_{is}\times f_{js})=0, \\ \dfrac{\sum_{s=1}^{n}\left(1-\left|f_{is}-f_{js}\right|\right)}{n} & if \quad \sum_{s=1}^{n} f_{js}\neq 0, \end{cases} \qquad (3\text{-}3)$$

Figure 4 shows that the concept-based model yields higher accuracy than the keyword-based model from one to ten query tests.

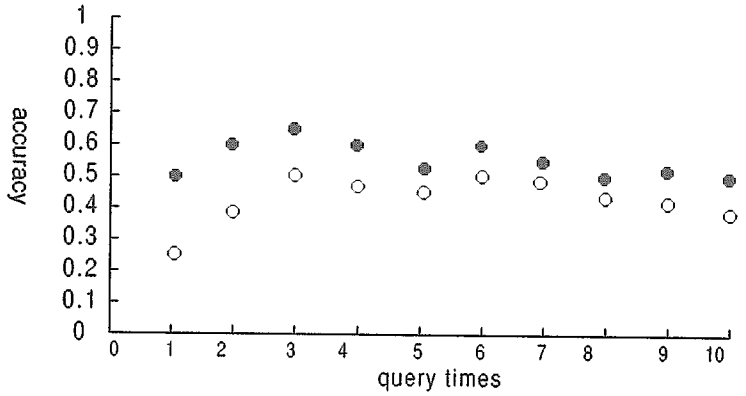


Figure 4: Concept model vs. keyword model w.r.t. different times of queries.

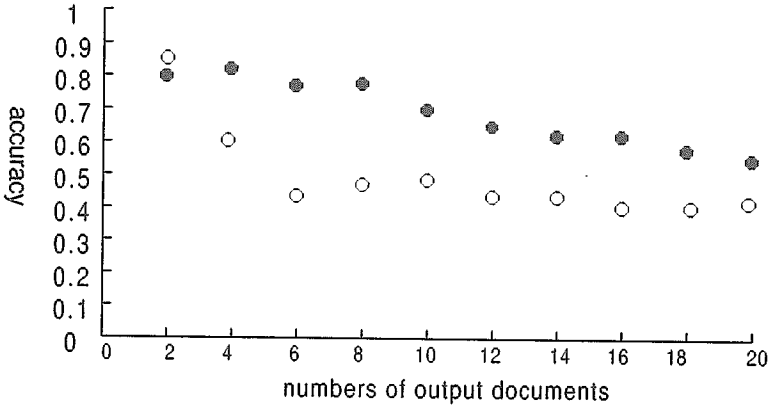


Figure 5: Concept model vs. keyword model w.r.t. numbers of output documents.

In addition, the proposed model produces higher accuracy, as shown in Figure 5, than the keyword -based model with respect to the different number of output documents from 4 to 20. The keyword model outperforms the concept-based model only when the number of output documents is two. This is because keyword-based model uses almost all the words occurring in the input document during relevance feedback and it is usual that the top-similarity two retrieved documents contain the most number of words in common and become the most similar documents.

4. Conclusion

In this paper, a fuzzy concept network particularly for Chinese textual retrieval is presented and implemented. There are several advantages provided by the proposed retrieval system. First is that both queries and documents are associated with concepts rather than specific keywords in the proposed system, therefore query expression becomes easier for end-users. The second advantage is the automatic computation of relation values rather than manual assignment, thus maki ng such concept-based retrieval practically useful in real applications. The third advantage is associated with the implementation of the proposed network with concept matrices, therefore document retrieval in the network becomes matrix computation which t urns out to simplify and speedup search procedure

On the other hand, the proposed retrieval system is incorporated with the capability of relevance feedback b the proposed conce pt-based relevant document retrieval module. Experimental results show that the proposed module achieves higher retrieval accuracy than a general keyword-based retrieval.

However the drawbacks with the proposed network retrieval are that the cost to handle frequent insertion of documents is high. The other drawback is associated with the implementation of concept matrices which will result in large space overhead. Further improvement with a matrix compression scheme can be in our future research direction.

Acknowledgement: This paper is partially supported by National Science Council, R. O. C. under the contract NSC-87-2213-E-009-087.

5. Reference

[1] S. M. Chen, J. Y. Wang, "Document retrieval using knowledg -based fuzzy information retrieval

techniques", *IEEE Transactions on Systems, Man and Cybernetics,* 25(5), 793-803, 1995.

[2] S. M. Chen, "Measures of similarity between vague sets", *Fuzzy Sets and Systems*, 74, 217- 223, 1995.

[3] L. K. Hyuan, Y. S. Song and K. Myung, "Similarit measure between fuzz sets and between elements", *Fuzzy Sets and Systems*, 62, 291-293, 1994.

[4] I. Itzkovich and L. W. Hawkes, "Fuzzy extension of inheritence hierarchies", *Fuzzy Sets and Systems*, 62, 143-153, 1994.

[5] G. J. Klir and B. Yuan, "Fuzzy sets and fuzzy logic theory and application", Englewood Cliffs, N.J, 1995.

[6] D. Lucarella and R. Morara, "FIRST: Fuzz information retrieval system", *Journal of Information Science,*17, 81-91, 1991.

[7] S. Miyamoto, "Information retrieval based on fuzz associations", *Fuzzy Sets and Systems*, 38, 191-205, 1990.

[8] C. P. Pappis and N. I. Karacapilidis, "A comparative assessment of measures of similarity of fuzz values", *Fuzzy Sets and Systems*, 56, 171-174, 1993.

[9] G. Salton, "Automatic Text Processing" : The Transformation, Anal ysis, and Retrieval of Information by Computer, Reading, MA: Addison-Wesley, 1989.

[10] Y. Y. Yang, "Autonmatic document classification", master thesis, National Tsin Hwa Univerity, Taiwan, 1992.