Acceleration of Building Thesaurus in Fuzzy Information Retrieval Using Relational products

Chang-Min Kim[a] and Yong-Gi Kim[b]

Dept. of Computer Science, Information and Communication Research Center
Gyeongsang National University
Chinju, Kyungnam, Korea
Tel:+82-591-751-5997 Fax:+82-591-762-1944
a.kimch@ailab.gsnu.ac.kr
b.ygkim@nongae.gsnu.ac.kr

Abstract

Fuzzy information retrieval which uses the concept of fuzzy relation is able to retrieve documents in the way based on not morphology but semantics, dissimilar to traditional information retrieval theories. Fuzzy information retrieval logically consists of three sets: the set of documents, the set of terms and the set of queries. It maintains a fuzzy relational matrix which describes the relationship between documents and terms and creates a thesaurus with fuzzy relational product. It also provides the user with documents which are relevant to his query. However, there are some problems on building a thesaurus with fuzzy relational product such that it has big time complexity and it uses fuzzy values to be processed with floating-point.

Actually, fuzzy values have to be expressed and processed with floating-point. However, floating-point operations have complex logics and make the system be slow. If it is possible to exchange fuzzy values with binary values, we could expect speeding up building the thesaurus. In addition, binary value expressions require just a bit of memory space, but floating-point expression needs couple of bytes.

In this study, we suggest a new method of building a thesaurus, which accelerates the operation of the system by pre-applying an *α-cut*. The experiments show the improvement of performance and reliability of the system.

Keyword: fuzzy information retrieval, fuzzy relational request, fuzzy relational products, thesaurus, *α-cut*

1. Introduction

One of the most fundamental notions in pure and applied sciences is the concept of a relation. Science has been described as the discovery of relations between objects, states and events. It is undeniable that the concept of a fuzzy relation has thoroughly enriched the applicability of this fundamental concept. In a fuzzy relation, one is considering objects that are related in some degree instead of dealing with related or non-related objects. The fuzzy relation has been used in many parts such as fuzzy information retrieval, medical diagnosis and approximate reasoning.[4].

Fuzzy information retrieval which uses the concept of fuzzy relation is able to retrieve documents in the way based on not morphology but semantics, dissimilar to traditional information retrieval theories[3]. Fuzzy information retrieval logically consists of three sets: the set of documents, the set of terms and the set of queries. It maintains a fuzzy relational matrix which describes the relationship between documents and terms and creates a thesaurus with fuzzy relational product. It also provides the user with documents which are relevant to his query. However, there are some problems on building a thesaurus with fuzzy relational product such that it has big time complexity or it uses floating-point to process fuzzy values.

In this paper, we study a way of building a thesaurus in fuzzy information retrieval system and a method for improving the performance of operation on building the thesaurus.

2. Fuzzy Information Retrieval model

The global structure of the fuzzy information retrieval model[6,8] which Kohout, Keravnou and Bandler proposed as an extension of a boolean retrieval model is depicted in Fig. 1. The input of the model consists of documents, thesaurus, fuzzy search requests and relational requests. The fuzzy search requests(FS-requests) are concerned with the questions related to the documents, and the relational requests(R-requests) are concerned with the questions related to the thesaurus. The output consists of relational output and fuzzy search output. FS-output is the answer to the FS-request, and is a list of references to documents; the R-output is the answer to an R-request and is a list of index terms[8].

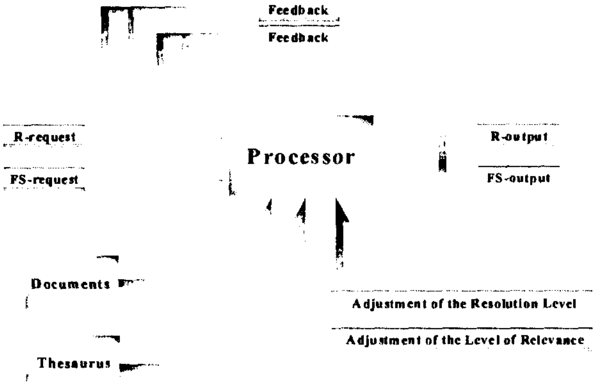

Fig. 1. Fuzzy Information Retrieval Model

2.1. Fuzzy Relational Products – BK-products

The fuzzy relational products which are based on the fuzzy information retrieval method are extended from binary relational products by Kohout and Bandler and they are used to build a thesaurus in fuzzy information retrieval system.

In fuzzy set theory, the words 'A fuzzy set *A* is a subset of a fuzzy set *B*' have a meaning as Eq. (1).

$$\begin{aligned} \tilde{A} \subset \tilde{B} &\Leftrightarrow \mu_{\tilde{A}} \le \mu_{\tilde{B}} \\ &\Leftrightarrow \forall x \in U,\ \mu_A(x) \le \mu_B(x) \end{aligned} \qquad (1)$$

The degree of the words 'A fuzzy set *A* is a subset of a fuzzy set *B*' can be described as Eq. (2).

$$\frac{1}{|U|}\sum(\mu_{\tilde{A}}(x) \to \mu_{\tilde{B}}(x)) \qquad (2)$$

It is the average of degree of the words 'A fuzzy set *A* is a subset of a fuzzy set *B*'. It is more flexible than the way taking a minimum value such as in crisp sets and so it is used in fuzzy set more generally. In Eq. (2), we can see that the implication operator is used. The implication operator, however, is implemented by various ways[1,5,7].

Let *R* be a fuzzy relation from fuzzy set *A* to fuzzy set *B* and let *S* be a fuzzy relation from fuzzy set *B* to fuzzy set *C*, fuzzy relational products of *R* and *S* have meanings on relationship of fuzzy set *A* and *C*.

$$(R \lhd S)_{ik} = \frac{1}{|B|}\sum(R_{ij} \to S_{jk}) \qquad (3)$$

$$(R \triangleright S)_{ik} = \frac{1}{|B|}\sum(R_{ij} \leftarrow S_{jk}) \qquad (4)$$

$$(R \square S)_{ik} = \frac{1}{|B|}\sum(R_{ij} \leftrightarrow S_{jk}) \qquad (5)$$

Let $a_i \in A$, $c_k \in C$. In Eq. (3), fuzzy triangular sub-product $(R \triangleleft S)_{ik}$ means the degree that a_i is contained into c_i In Eq. (4), fuzzy triangular super-product $(R \triangleright S)_{ik}$ means the degree that a_i contains c_i. In Eq. (5), fuzzy square product $(R \square S)_{ik}$ means the degree that a_i is similar to c_i. Fuzzy relational products like above can be use to build a thesaurus in fuzzy information retrieval[6,8].

2.2. Thesaurus and Relational Request

The R-request[1,2,5] shows a relation between a given term and the others using a thesaurus constructed. A thesaurus, therefore, must be produced in advance to process R-request.

Given a fuzzy relation R (R : documents × terms), building a thesaurus of terms on fuzzy relation R follows the next step. i) Yields a result relation using the fuzzy triangular sub-product of input relation R and its transpositive relation R^T. ii) Converts the result relation to a binary relation by applying *α-cut*. iii) Yields a thesaurus by applying a Hasse diagram.

2.3. Applying *α-cut*

α-cut is one of the most general methods to convert a fuzzy set to a crisp set. Given *α-cut*= c ($0.0 \le c \le 1.0$), if a input value is greater than or equals to c, it is converted into 1, and if a input value is less than to c, it is converted into 0.

2.4. Building Thesaurus with Hasse Diagram

Hasse Diagram is a method of representing a relationship to a diagram. If a binary relation is given, drawing a Hasse diagram from the relation consists of following steps.

① Prepare for a binary relation as a input relation.

② Remove all self-loops.

③ Remove all transitivity lines.

④ Arrange all lines so that they are pointing up.

2.5. Problems in traditional method

Set the number of terms to T_N and the number of documents to D_N, the operation on building a thesaurus has the time complexity to be proportioned to $D_N^2 \times T_N$. Fuzzy values are real numbers and they are implemented with floating-point expressions on current computer architectures. Floating-point operations, however, have very complex logic and request larger memory space. Therefore, the fuzzy relational products which have higher time complexity and use floating-point operations is very weak in capability of processing.

3. Pre-Applying *α-cut*

In this paper, we suggest a new method which improves the weakness in building a thesaurus by applying *α-cut* in advance. The method saves memory spaces and improves the speed of processing by converting floating-point expression to binary logical expression.

If a fuzzy input relation is given, convert it into a binary input relation by applying an *α-cut*. The binary relation products suggested in the paper are represented in Eq. (6)(7)(8).

$$(R \triangleleft S)_{ik} = \begin{cases} 0, if \dfrac{1}{|B| \times A}\sum(R_{ij} \rightarrow S_{jk}) < 1 \\ 1, otherwise \end{cases} \qquad (6)$$

$$(R \triangleright S)_{ik} = \begin{cases} 0, if \dfrac{1}{|B| \times A}\sum(R_{ij} \leftarrow S_{jk}) < 1 \\ 1, otherwise \end{cases} \qquad (7)$$

$$(R \square S)_{ik} = \begin{cases} 0, if \dfrac{1}{|B| \times A}\sum(R_{ij} \leftrightarrow S_{jk}) < 1 \\ 1, otherwise \end{cases} \qquad (8)$$

A is the value to adjust processed result on the property of relation and *α-cut*. In this paper, it is set using Eq. (9).

$$A = \alpha_{cut} + (1 - \alpha_{cut})^2 \times 0.8 \times (1 - \frac{4.5}{\sqrt{|B|}}) \qquad (9)$$

Even if fuzzy implication operators are implemented to dozens of ways[1,7], binary implication operators have single meaning such as Eq. (10)(11)(12).

$$a \rightarrow b = \neg a \vee b \tag{10}$$
$$a \leftarrow b = a \vee \neg b \tag{11}$$
$$a \leftrightarrow b = (\neg a \vee b) \wedge (a \vee \neg b) \tag{12}$$

4. Evaluation

The suggested method can be evaluated in view of efficiency and reliability. Efficiency is inspected based on memory space and speed of processing.

4.1. Efficiency of the method

Set the number of terms to T_N and the number of documents to D_N, the operation on building a thesaurus has the time complexity to be proportioned to $D_N^2 \times T_N$. Therefore, if it is possible to convert fuzzy values into binary values, the time of the process can be reduced conspicuously. Table 1 shows the result of experiment comparing how many implication operators(→) can be processed on two workstations.

As it is easily seen from Table 1, binary implication operator is 105 times as fast as fuzzy implication operator. There are two reasons which are that binary logic is simpler than floating-point logic and a vector process is applied to 64 terms at the same time in case of the 64-bit machine.

Table 1. Speed on processes of implication operator

Machine	Processes per a second		Rate of speed (X/Φ)
	Fuzzy implication operator(Φ)	Crisp implication operator(X)	
A(64 bits)	0.476×10^7	0.457×10^{12}	0.960×10^5
B(32 bits)	0.149×10^7	0.604×10^{10}	0.405×10^4

- **Fuzzy implication operator: min(1,b/a)[1,5,7]**
- **Crisp implication operator: ~avb**

Given a fuzzy relation from a set of documents that the number of members is D_N to a set of terms with 320 members, Fig.1 shows the change of required time for processing fuzzy triangular sub-product($R^T \lhd R$) depending on D_N. We can realize that the traditional method requires about 6 times as large time as the suggested method. The Suggested method, however, could not reach the expected level. The most important reason of the result is that considerable volume of processing time is assigned to loops in the program.

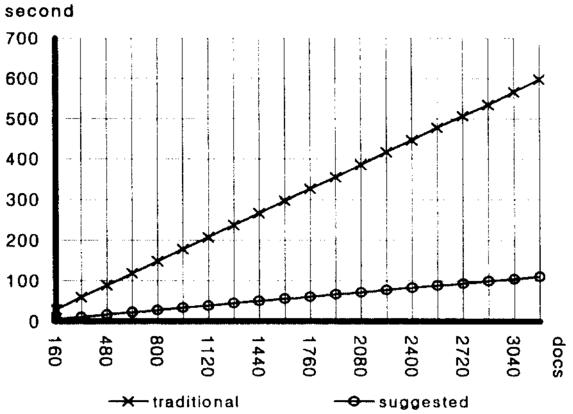

Fig. 1 time of building thesauruses

Let's compare the two methods in side of memory space on 32-bit machine. Given a machine using 4 bytes to represent a real value, the suggested method requires just one bit to express a term, but the traditional method needs 4 bytes, so it comes to reduce memory space to 1/32.

4.2. Reliability of the method

The suggested method converts fuzzy values into binary values to improve the speed of building a thesaurus in the beginning but it causes to lose the volume of information. It means that the thesaurus built by the suggested method is not more reliable than by the traditional method.

In this chapter, we inspect and compare statistically two thesauruses built by the two different methods, and evaluate similarity between relations so that we know how much information is lost after the processes.

4.2.1. Similarity of two fuzzy relations

In this paper, we suggest a method to measure similarity between two fuzzy relations.

As a arbitrary fuzzy set, $\widetilde{X}$ and $x_1 \in X$, $x_2 \in \widetilde{X}$, $\overline{x_1} = \mu_{\widetilde{X}}(x_1)$, $\overline{x_2} = \mu_{\widetilde{X}}(x_2)$ are given, the similarity of x_1 and x_2 can be defined as Eq. (13).

$$S_{mem}(x_1, x_2) = \begin{cases} 1-(\overline{x_1}-\overline{x_2}), if\ \overline{x_1} \geq \overline{x_2} \\ 1-(\overline{x_2}-\overline{x_1}), if\ \overline{x_1} < \overline{x_2} \end{cases} \quad (13)$$

As arbitrary fuzzy relations, $R^{\alpha} : \widetilde{X} \times \widetilde{Y}$, $R^{\beta} : \widetilde{X} \times \widetilde{Y}$ are given, the similarity of R^{α} and R^{β} can be defined like Eq. (14).

$$S_{rel}(R^{\alpha}, R^{\beta}) = \frac{1}{|X| \times |Y|} \sum_{i=1}^{|X|} \sum_{j=1}^{|Y|} S_{mem}(R^{\alpha}{}_{ij}, R^{\beta}{}_{ij}) \quad (14)$$

4.2.2. Plan of the experiment

The experiment follows the next steps. First of all, creates a fuzzy relation R whose size is $N \times M$ using a random number generator, creates R^T which is the transpositive relation of R and yields a result relation R_I with fuzzy triangular super-product ($R^T \triangleright R$). Additively, the result of the traditional method, R_T is figured out by applying *α-cut* to R_I. In the other side, the result of the suggested method, R_S is yielded from R with the method.

How can we know the reliability of results by two methods. There is a considerable point that the fuzzy relation, R_I preserves information which is nearly close to real status. Therefore, the reliability of the traditional method is known with similarity between R_I and R_T like Eq. (15) and the reliability of the suggested method is known with similarity between R_I and R_S like Eq. (16).

$$S_O = S_{rel}(R_I, R_T) \quad (15)$$

$$S_S = S_{rel}(R_I, R_S) \quad (16)$$

4.2.3. Comparison in view of reliability

Fig. 2 shows the results of the experiment with a fuzzy input relation size of 3200 × 320. The fuzzy input relation is created with a random number generator and consists of real numbers ranged from 0 to 1. S_T is a curved line that represents the change with the traditional method, S_S is a curved line that represents the change with the suggested method and they are decreased, *α-cut* increased. In Fig. 2, the line of S_T is located in upper side to the line of S_S. It means that volume of information lost is larger in the suggested method than in the traditional method.

In Fig. 2, a phenomenon which S_S is suddenly increased near by *α-cut*=1 is detected. As *α-cut* is close to 1, the fuzzy input relation becomes the null relation nearly and it cause to uniform result relation and to lost characteristics of the fuzzy input relation. Adjustment in Fig. 2 is the same as that in Eq. (9) and it has range form 0 to 1. It is designed to revise the loss of information.

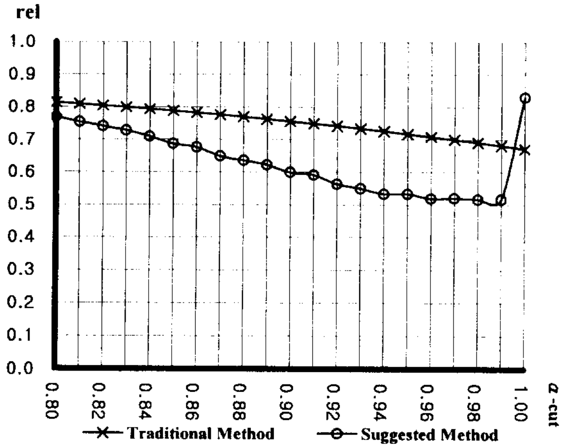

Fig. 2 Comparison of reliability

5.Conclusion

Fuzzy information retrieval which uses the concept of fuzzy relation is able to retrieve documents in the way based on not morphology but semantics, dissimilar to traditional information retrieval theories.

It maintains a fuzzy relational matrix which describes the relationship between documents and terms and creates a thesaurus with fuzzy relational product. However, there are some problems on building a thesaurus with fuzzy relational product such that it has big time complexity and it uses floating-point to process fuzzy values.

In this paper, we suggest and evaluate an *α-cut* pre-applying method to build a thesaurus in fuzzy information retrieval system. The suggested method has two benefits. One is that it has a capability as 6 times as the traditional method. The other is that it is able to save more memory spaces. In the side of reliability, however, the suggested method is not as good as the traditional method. The suggested method which is applied in beginning, loses more information than the traditional method reasonably. In conclusion, if one needs a fuzzy information retrieval system which requires not many documents and terms, it is reasonable to select the traditional method. However, if one needs a fuzzy information retrieval system which requires so many documents and terms, the suggested method is better choice.

Reference

[1] Bandler, W. and Kohout, L. J., "Fuzzy power sets and fuzzy implication operators", in Fuzzy sets and systems edited by Wang, P. P. and Chang, S. K., plenum press, 1980

[2] Bandler, W. and Kohout, L. J., "Fuzzy products as a tool for analysis and synthesis of the behaviour of complex natural and artificial systems", in Theory and Applications to Policy Analysis and Information Systems edited by Wang, P. P., Plenum Press, 1980

[3] Frakes, William B. and Baeza-Yates, Ricardo, Information Retrieval Data Structures & Algorithms, Prentice Hall, 1992

[4] Kerre, E. E., "A walk through fuzzy relations and their application to information retrieval, medical diagnosis and expert systems", Elservier Science Pub., 1992

[5] Kim, Yong-Gi and Kohout, L. J., "Comparison of Fuzzy Implication Operators by means of Weighting Strategy in Resolution Based Automated Reasoning", SAC '92 Proceedings, pages, ACM Symposium on Applied Computing Kansas City, MO, 1992

[6] Kohout, L. J. and Bandler, W., "Relational-Product Architectures for Information Processing", Information Science, Elsevier Science Publishing, 1985

[7] Kohout, L. J. and Bandler, W., "Semantics of implication operators and fuzzy relational products", International Journal of Man-Machine Studies 12, 1980

[8] Kohout, L. J., Keravnou, E. and Bandler, W., "Automatic Documentary Information Retrieval by means of Fuzzy Relational Products", in Fuzzy Sets in Decision Analysis by Gaines, B. R., Zadeh L. A. and Zimmermann, H. J., pages 308-404, North-Holland, Amsterdam, 1984

[9] Pinkava, V., “Fuzzification of binary and finite multivalued calculi, Int. J. Man-Machine Studies, 8, 1976

[10] Rijsbergen, C. J. van, Information Retrieval, 2nd ed, butterworths, 1979