World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:4, No:3, 2010

Applications of Rough Set Decompositions in Information Retrieval

Chen Wu, Xiaohua Hu

***Abstract*—**This paper proposes rough set models with three different level knowledge granules in incomplete information system under tolerance relation by similarity between objects according to their attribute values. Through introducing dominance relation on the discourse to decompose similarity classes into three subclasses: little better subclass, little worse subclass and vague subclass, it dismantles lower and upper approximations into three components. By using these components, retrieving information to find naturally hierarchical expansions to queries and constructing answers to elaborative queries can be effective. It illustrates the approach in applying rough set models in the design of information retrieval system to access different granular expanded documents. The proposed method enhances rough set model application in the flexibility of expansions and elaborative queries in information retrieval.

***Keywords*—**Incomplete information system, Rough set model, tolerance relation, dominance relation, approximation, decomposition, elaborative query.

I. Introduction

INFORMATION retrieval (IR) is one of the four modules: information retrieval (IR), entity recognition (ER), information extraction (IE), and text mining (TM) in a text mining system. IR retrieves documents relevant to certain interested topics. ER identifies concepts for further information extraction and text mining. IE extracts facts involving two or more entities. TM performs either traditional text mining tasks including text clustering, text classification, topic detection, and multi-document summary, or more advanced hypothesis generation and discovery. Due to the complexity of text mining, many new approaches such as probabilistic, Bayesian, or Markov models have been introduced [1,8,9].

Rough set theory [7,11] has been recently applied in Intelligent Systems, Machine Learning, Pattern Recognition, Data Mining, Decision Making and etc[1,3,5,6,9]. In an incomplete information system, an indiscernibility relation between objects according to definitive attribute values of them may no longer be formed because of existing null values(not non-applicable) to some attributes of certain objects. How to deal with it properly? Two approaches are always introduced by researchers [3,4,5,7,9]. One is to filling them out with high frequently occurring values by statistics. The other is to extend original RST model to facilitate incomplete systems to be processed directly.

Because it is hard to find the real occurrence of values in the null positions in filling out by statistics or other high computation, directly handling becomes an hot topic. For example, Kryszkiewicz M. generalized equivalence relation into tolerance relation possessing reflexivity and symmetry [6].Similarity relation, valued tolerance relation, and etc. are also introduced by other experts[4,7,11]. In information retrieval, approaches based on rough set model are proposed [2]. But a lack of retrieval diversity or a limitation of completeness becomes restricts.

Implementing diverse information retrieval in text mining in incomplete information system is more sophisticated than that in complete one because null values have to be processed and tolerance but not equivalence relation may only be created. At first, similarity measure between numerical or non numerical attribute values of two objects is set up. Then the similarity measure between objects is formed. Thirdly, a tolerance relation based on object similarity is established. A tolerance relation is not always an equivalence relation which at least has transitivity, so a partition might not be formed, but a collection of similar classes can be constructed. With dominance relation referring to attribute values of objects, similar classes can be further decomposed into three subclasses: little better sub-classes, little worse subclasses and vague subclasses. A query naturally includes a term subset X. Requirements of queries differ from each other according to their semantics. For certain queries, the decomposed lower or upper approximation by dominance relation can meet the needs more similar or precise. We show how to get expansions and to retrieve information by decompositions for elaborative queries.

The paper is organized as follows. Section 2 defines a similarity measurement for determining similarity relationship between two objects through attribute values. Section 3 reviews concepts of document-term matrix representation. Section4 suggests three granules using maximally compatible classes as primitives. Section 5 introduces ideas and principles of new approximations and decompositions under dominance relation based on the granules. Section 6 demonstrates expansions at different levels for a given query and implements elaborative queries through different granules proposed by us. Section 7 concludes our study.

Chen Wu is with Jiangsu University of Science and Technology, Zhenjiang City, Jiangsu Province,P.R.China (corresponding author to provide phone:01-0511-84401368; e-mail: wuchenzj@sina.com).

Xiaohua Hu is with Drexel University, Philadelphia, PA 19104 USA. (e-mail: thu@cis.drexel.edu).

II. Basic Concepts

As in [6], an incomplete information system (IIS for short) is a quadruple $S=(U,AT,V,f)$, and a tolerance relation derived by $A\subseteq AT$, named $SIM(A)$, is defined as in [6]. The tolerance class $S_A(x)$ for $x\in U$ is probably the biggest subset in which each element has tolerance relation $SIM(A)$ with x and can not be distinguished with x. All elements in $S_A(x)$ are tolerant with x, but anyone can not make sure whether or not two elements in $S_A(x)$ are compatible except one of them is x.

Different from traditional document representation approach, a given set of text documents is represented by an incomplete information system. Each document in a corpus is expressed by a term vector in vector space model (VSM). Traditionally, weights of those terms not occurring in a document are set zero. But some potentially relevant information would be lost due to such an unreasonable representation. To complement this shortage and not to degrade retrieval performance, incomplete information system is preferred to be used to describe data set. From the prospective of information system, such a consideration is natural and rational.

Instead of assigning zero to weights of those terms absent in a document, their weights are considered missing. Based on this idea, an approach of representing documents as incomplete term vectors is proposed. In this way, an incomplete information system with some missing values is constructed. Information loss can be avoided and therefore retrieval quality can be promoted. Suppose we have m documents and n different terms in a corpus. Each document is expressed by an n-dimensional term vector. Every different term occurred in the corpus is viewed as a dimension in term vector.

In order to apply rough set model in general, some real-valued weights have to be discretized. The most often used method is using "0" and "1" to represent the weights. But, the term vectors cannot reflect the extent of how frequent it is for each term and thus the computed similarity measure of documents can not well reflect the actual similarity between documents. To overcome this kind of information loss, the representation of term vectors here is not discretized in binary formation but somewhat in fuzzy form. RST models are then used to extract hidden and associative terms which convey an idea for text retrieval in the incomplete information system. The missing weights of extracted terms of a document do not need filled. Documents in the original text dataset are expanded and the ordinary smoothing work is also got rid of.

III. Similarity Measurement

Let $a\in A\subseteq AT$ be an attribute of the system. $R_a\subseteq V_a^2$. If for any $v\in V_a$, $(v,v)\in R_a$, then R_a is reflexive. For any $v_i\in V_a$ and $v_j\in V_a$, if $(v_i,v_j)\in R_a$ and $(v_j,v_i)\in R_a$, then R_a is called symmetric. Objects x, y are called similar with respect to a, if $(a(x),a(y))\in R_a$. $q_a:V_a\times V_a\rightarrow[0,1]$ is a measurement function to compute similarity between two attribute values v_i and v_j in V_a. q_a is symmetry, i.e., $q_a(v_i,v_j)$=$q_a(v_j,v_i)$, and is capable of processing null values, i.e., v_i or v_j may be *. If one of them is *, then $q_a(v_i,v_j)$=1. Assume that $\lambda_a\in[0,1]$ is a threshold for a, which is given by domain experts and supposed to be a minimal value for similarity between v_i and v_j. If $q_a(v_i,v_j)\geq\lambda_a$, then v_i and v_j are said to be similar, otherwise dissimilar.

(1) If $a\in A$ is continuous and numerical, and any one of v_i, v_j is not *, then q_a (v_i , v_j)= $1-|v_i-v_j|$ / (sup(V_a)-inf(V_a)), where sup(V_a) is the superior value of V_a and inf(V_a) is the inferior value of V_a.

(2) If a is discrete, numerical, finite and ordered, and anyone of v_s, v_t is not *, then q_a (v_s , v_t)= $1-|s-t|/(k-1)$, where V_a = $\{v_1, v_2, \ldots, v_s, \ldots, v_t, \ldots, v_k\}$, ($k=card(V_a)$ and $v_1<v_2<\ldots<v_s<\ldots<v_t<\ldots<v_k$.

(3) If v_s is * or v_t is * or both of them are * for $a\in A$ is numerical or not, then $q_a(v_s, v_t)$ is 1.

(4) If $a\in A$ is non-numerical but symbolic, then $q_a(v_s,v_t)$ is given by domain experts according to different semantics of attributes. Referring to [4], an aggregative similarity measure function q_A as similarity degree between two objects on attribute subset can also be defined.

Let $A\subseteq AT$. The similarity degree between x and y is defined by $q_A(x,y)=f(A,q_a(a(x),a(y)))$. If $q_A(x,y)=1$, then x is similar to y, otherwise, they are dissimilar. f depends on domain knowledge or domain experts. If all attributes in A are numerical attributes, then one of following formula can be used.

$$q_A(x,y)=\begin{cases}1, \forall a\in A, q_a(a(x),a(y))\geq\lambda_a\\ \quad 0, otherwise\end{cases}$$

$$q_A(x,y)=\begin{cases}1, \forall a\in B, q_a(a(x),a(y))\geq\lambda_a, B\subseteq A,\\ |B|/|A|\geq 1/2\\ \quad 0, otherwise\end{cases}$$

$$q_A(x,y)=\begin{cases}1, \sum_{a\in A} q_a(a(x),a(y))\geq\Delta_A\\ \quad 0, otherwise\end{cases}$$

where λ_a is a threshold for attribute a and Δ_A is an aggregating threshold for attribute subset A. After gaining a definition of similarity degree between objects, a similarity relation on U can be obtained:

$$SIM_A = \{(x,y) \mid q_A(x,y)=1\}.$$

Dominance relation is defined by $D_A = \{(x,y) \mid xP_A y\}$, where $xP_A y$ if , and only if $xP_a y$ for each a in A. If a is numerical attribute, then $xP_a y$ if and only if $a(x)$ dominates $a(y)$, or both $a(x)$ and $a(y)$ are not * or any or both of $a(x)$ and $a(y)$ are *. If a is a non-numerical attribute, then $xP_a y$ is determined by domain knowledge or experts. If any of them is *, then dominance relation always holds between them.

If a partially ordered relation exists in a non-numerical attribute value space, then a dominance relation can be easily established. For example, $\leq, \geq$ may be a dominance relation on numerical value attribute, while $\supseteq, \subseteq$ may be a dominance relation in any collection of subsets on a set. A dominance relation has reflexivity, transitivity but not necessarily symmetry. We will build a dominance relation by attribute value comparison not by object priority to combine with a tolerance relation to make decompositions in our paper in section 5.

IV. Primitive and Non-primitive Granules

Since tolerance relation SIM_A derived by $A \subseteq AT$ is also a compatible relation actually, we use another symbol $CSIM_A$ to substitute for SIM_A i.e. $CSIM_A = SIM_A$. However, its complete covering, denoted by $U/CSIM_A$, is different from U/SIM_A. $U/CSIM_A$ is defined in the following:

$$U/CSIM_A = \{X \subseteq U : X \times X \subseteq CSIM_A,$$
$$\forall x(x \in U \wedge x \notin X \rightarrow (X \cup \{x\})^2 \not\subseteq CSIM_A)\}.$$

The difference between them is that $U/CSIM_A$ is made up from maximal compatible classes whereas U/SIM_A is consisted of tolerance classes. $U/CSIM_A$ and U/SIM_A may not form partitions on U.Two objects in a maximal compatible class always have compatibility with $CSIM_A$. Maximal compatible class is a compatible class which could not be included in another else compatible class. If adding an extra object to a maximal compatible class, the compatibility of its maximum will be broken. Elements of $U/CSIM_A$ are considered to be primitive granules by us. Two other kinds of granules for $x \in U$ are formed from $U/CSIM_A$:

$$SU_A(x) = \cup X(X \in U/CSIM_A \wedge x \in X) \;;$$
$$SL_A(x) = \cap X(X \in U/CSIM_A \wedge x \in X) \cdot$$

Two arbitrary elements in $SL_A(x)$ are still compatible, but it is not true in $SU_A(x)$. $\{SU_A(x) \mid x \in U\}$, $\{SL_A(x) \mid x \in U\}$ and $U/CISIM_A$ are three knowledge expression systems [3,11] at different levels. Three lower and upper approximations may also be respectively obtained:

$$\overline{SU}_A(X)=\{x \in U : SU_A(X) \cap X \neq \emptyset\};$$
$$\overline{SL}_A(X)=\{x \in U : SL_A(X) \cap X \neq \emptyset\};$$
$$\overline{SC}_A(X)=\{x \in U : Z \cap X \neq \emptyset, Z \in U/CISIM_A\};$$
$$\underline{SU}_A(X)=\{x \in U : SU_A(X) \subseteq X\};$$
$$\underline{SL}_A(X)=\{x \in U : SL_A(X) \subseteq X\};$$
$$\underline{SC}_A(X)=\{x \in U : x \in Z, Z \subseteq X,\ Z \in U/CISIM_A\}$$

V. Decompositions and Approximation under Dominance Relation

By dominance relation, documents corpus D can be divided into two subsets, denoted by $D_{\mathrm{p}}{}^{+}(d_i)$ and $D_p{}^{-}(d_i)$ respectively with respect to d_i. $D_{\mathrm{p}}{}^{+}(d_i)$ includes all elements dominating d_i and $D_p{}^{-}(d_i)$ includes all elements dominated by d_i. According to a theorem 3 [2], we have $S_A(d_i) = SU_A(d_i)$. We can use $S_A(d_i)$ or $SU_A(d_i)$ freely and equivalently. Similar class $S_A(d_i)$ may be decomposed into three subclasses[4]: positively similar subclass $S_A^+(d_i)$, negatively similar subclass $S_A^-(d_i)$, and purely similar subclass $S_A^0(d_i)$. $S_A^+(d_i) = S_A(d_i) \cap D_P^+(d_i)$ represents a collection of elements similar to and a little bit better than d_i . $S_A^-(d_i) = S_A(d_i) \cap D_P^-(d_i)$ denotes a collection of elements similar to and a little bit worse than d_i . $S_A^0(d_i)$ £ $S_A\ddot{}(d_i) - \mathrm{D}_{\mathrm{P}}^+(d_i) - \mathrm{D}_{\mathrm{P}}^-(d_i)) \cup \{d_i\}$ expresses a collection of elements purely similar to d_i . So $S_A(d_i) = S_A^+(d_i) \cup S_A^-(d_i) \cup S_A^0(d_i)$.

In this way, $SL_A(d_i)$ may also be divided into three overlapped subclasses: $SL_A^+(d_i)$, $SL_A^-(d_i)$, $SL_A^0(d_i)$, and $SC_A(d_i)$ into three overlapped subclasses: $SC_A^+(d_i)$, $SC_A^-(d_i)$, $SC_A^0(d_i)$. We also have:

$$SC_A(d_i) = SC_A^+(d_i) \cup SC_A^-(d_i) \cup SC_A^0(d_i);$$
$$SL_A(d_i) = SL_A^+(d_i) \cup SL_A^-(d_i) \cup SL_A^0(d_i) \cdot$$

According to the three subclasses in the decomposition of $S_A(d_i)$, we may obtain three groups of upper approximation, lower approximation and boundary subsets for any subset $X \subseteq D$.

For $SU_A^+(d_i)$, we define

$$\underline{SU_A}^+(X) = \cup SU_A^+(d_i)\,(\,SU_A^+(d_i) \subseteq X\,);$$

$$\overline{SU_A}^+(X) = \cup SU_A^+(d_i)\,(\,SU_A^+(d_i) \cap X \neq \varnothing\,);$$

$$BndSU_A^+(X) = \overline{SU_A}^+(X) - \underline{SU_A}^+(X).$$

Similarly, for $SU_A^-(d_i)$ and $SU_A^0(d_i)$ we can also define $\underline{SU_A}^-(X)$, $\overline{SU_A}^-(X)$, $BndSU_A^-(X)$ and $\underline{SU_A}^0(X)$, $\overline{SU_A}^0(X), BndSU_A^0(X)$. According to [5], we have:

$$\underline{SU}_A(X) = \underline{SU_A}^+(X) \cap \underline{SU_A}^-(X) \cap \underline{SU_A}^0(X);$$

$$\overline{SU}_A(X) = \overline{SU_A}^+(X) \cup \overline{SU_A}^-(X) \cup \overline{SU_A}^0(X).$$

For $SC_A(d_i)$ and $SL_A(d_i)$, we can also give definitions of the following denotations: $\underline{SC}_A^+(X)$, $\overline{SC}_A^+(X)$, $BndSC_A^+(X)$, $\underline{SC}_A^-(X)$, $\overline{SC}_A^-(X)$, $BndSC_A^-(X)$, $\underline{SC}_A^0(X)$, $\overline{SC}_A^0(X)$, $BndSC_A^0(X)$ $\underline{SL}_A^+(X)$, $\overline{SL}_A^+(X)$, $BndSL_A^+(X)$, $\underline{SL}_A^-(X)$, $\overline{SL}_A^-(X)$, $BndSL_A^-(X)$, $\underline{SL}_A^0(X)$, $\overline{SL}_A^0(X)$, $BndSL_A^0(X)$. Similarly, it is not difficult to prove the following formula:

$$\underline{SC}_A(X) = \underline{SC}_A^+(X) \cap \underline{SC}_A^-(X) \cap \underline{SC}_A^0(X);$$

$$\overline{SC}_A(X) = \overline{SC}_A^+(X) \cup \overline{SC}_A^-(X) \cup \overline{SC}_A^0(X);$$

$$\underline{SL}_A(X) = \underline{SL}_A^+(X) \cap \underline{SL}_A^-(X) \cap \underline{SL}_A^0(X);$$

$$\overline{SL}_A(X) = \overline{SL}_A^+(X) \cup \overline{SL}_A^-(X) \cup \overline{SL}_A^0(X).$$

VI. Similarity Expansions and Elaborative Queries Using Decompositions Measurement

A document corpus can be represented by $S = (U, AT, V, f)$, where $U = D$ is the set of documents; each document is an object in D; $AT = TM \cup \{topic\}$ and TM is a set of total terms occurring in documents; and $topic$ is the decision attribute, i.e., class labels of documents. V, f are defined similarly in a information system. So $S = (D, TM \cup \{topic\}, V, f)$. As an example, let $U = D$ $=\{d_1, d_2, \ldots, d_{13}\}$ be a corpus, $TM = \{a,b,c,d\}$, $topic = \varnothing$. The term-documentary matrix are discretized as in Table 1. "*" represents zero or an ambiguity. Integers represent the regularized tf-idf weights or occurrence times.

Table I An IIS for Document representation

D	a	b	c	d	D	a	b	c	d
d1	12	22	13	t	d8	2	10	*	m
d2	14	22	*	t	d9	*	*	4	m
d_3	15	*	13	*	d_{10}	5	*	4	*m*
d_4	*	20	16	*t*	d_{11}	6	10	*	*b*
d_5	18	*	18	*s*	d_{12}	16	22	*	*h*
d6	18	27	18	s	d13	16	*	20	*
d7	2	10	4	*					

Attribute d expresses a fuzzy result. Its role is to deliberatively explain how to implement a non-binary discretization and to achieve the target of document expansions. An associative matrix is gained based on similarity measure. It is essential to find out primitives. But here it is omitted.

Table II Class $SU_A(d_i)$ and its decompositions

D	SU_A	SU_A^+	SU_A^-	SU_A^0
d1	d1-d4	d1-d3	d1	d1,d4
d2	d1-d5,d13	d2,d3, d5,d13	d2,d1,d4	d2
d3	d1-d4,d12	d3,d4, d12	d1-d3	d3
d4	d1-d5,d13	d2,d4, d5,d13	d3,d4	d4,d1
d5	d2,d4-d6, d13	d5	d2,d4,d5	d5,d6,d13
d6	d6,d5,d13	d6	d6	d5,d6,d13
d7	d7-d11	d7,d10,d11	d7	d7,d8,d9
d8	d7-d10	d8,d10	d8	d7,d8,d9
d9	d7-d10	d9	d9	d7-d10
d10	d7-d10	d7,d8, d10	d7,d8, d10	d9,d10
d11	d7,d11	d11	d11,d7	d11
d12	d12,d3,d13	d12	d12,d3	d12,d13
d13	d13,d2,d4-d6,d12	d13	d13,d2, d4	d5,d6, d12,d13

Similarity measure as in Section 3 is calculated as follows. Let λ_a =0.9, λ_b =0.85, λ_c =0.8, and $q_d("t","s")$=1, $q_d("b","h")$=1, and $q_d(v,v)$=1 for v in $\{"t","s","m","b","h"\}$, and $q_A(v_1,v_2)$=0 for other pairs (v_1,v_2), where v_1,v_2 are all in $\{"t", "s", "m", "b", "h"\}$. "t" stands for "tiny", "s" for "small", "m" for "medium", "b" for "big" and "h" for "huge". $"t"<"s"< "m" <"b"<"h"$ is supposed to be a dominance relation on Vd. λ_d =1. Δ_A =3.6 for A={a, b, c, d}. SIMA= $\{(x,y) \mid q_A(x,y)=1\}$.

Three different granules $SU_A(d_i)$, $SC_A(d_i)$, $SL_A(d_i)$ are counted at three levels. By natural dominance relation, three decompositions of granules are shown in Table 2, Table 3 and Table 4 respectively. Table 3 supplies multiple components.

1) Different expansions to a given query

Similar to neighborhood system, three different upper and lower approximations of documents are viewed as different neighborhoods. But they are naturally generated from incomplete information systems, so they are not very theoretical. The topic discussed here can be regarded as an application of neighborhood system. For each d_i, $\{d_i\} \subseteq SL_A(d_i) \subseteq SC_A(d_i) \subseteq SU_A(d_i)$ and $d_i \in H \in \{SL_A^+(d_i), SL_A^-(d_i), SL_A^0(d_i), SC_A^+(d_i), SC_A^-(d_i), SC_A^0(d_i), SU_A^+(d_i), SU_A^-(d_i), SU_A^0(d_i)\}$.

(1) Use $SL_A(d_i)$, $SL_A^+(d_i)$, $SL_A^-(d_i)$, $SL_A^0(d_i)$ to reach expansions. Let the query be: retrieve documents that satisfy: '((a=18)&(d="s")) or ((b=10)&(d="m"))'.The initial retrieval result is X={d5,d6,d7,d8,d9,d10,d13} under inner condition. Then, expansions to the senses of $SL_A(d_i)$, $SL_A^+(d_i)$, $SL_A^-(d_i)$, $SL_A^0(d_i)$ are the same:

$$\cup_{l=5}^{10} SL_A(d_l)\cup SL_A(d_{13}) = \cup_{l=5}^{10} SL_A^+(d_l)\cup SL_A^+(d_{13})$$

$$= \cup_{l=5}^{10} SL_A^-(d_l)\cup SL_A^-(d_{13}) = \cup_{l=5}^{10} SL_A^0(d_l)\cup SL_A^0(d_{13})$$

={d5,d6,d7,d8,d9,d10,d13}.

(2) Apply $SC_A(d_i)$ or its decompositions $SC_A^+(d_i)$, $SC_A^-(d_i)$, $SC_A^0(d_i)$ to gain expansions to the same query. The initial retrieval result is still X . Because many choices exist, expansions can be obtained alternatively. For $SC_A(d_5)$, two options denoted by $SC_A^{'}(d_5)$ ={d2,d4,d5,d13} and $SC_A^{''}(d_5)$ = {d5,d6,d13} respectively. Two alternatives for d_7 are:

$SC_A^{'}(d_7)$ ={d7,d8,d9,d10}, $SC_A^{''}(d_7)$ ={d7,d11}.

Meanwhile, there are three candidates for d13 to choose for $SC_A(d_{13})$:

$SC_A^{'}(d_{13})$ ={d2,d4,d5,d13}, $SC_A^{''}(d_{13})$ ={d5,d6,d13},

$SC_A^{'''}(d_{13})$ ={d12,d13}.

So many expansions are available. Here some are given as examples:

{d2,d4,d5,d6,d7,d8,d9,d10,d13};
{d5,d6,d7,d8,d9,d10,d13};
{d5,d6,d7,d8,d9,d10,d12,d13};
{d2,d4,d5,d6,d7,d8,d9,d10,d12,d13};
{d2,d4,d5,d6,d7,d8,d9,d10,d11,d13};
{d2,d4,d5,d6,d7,d8,d9,d10,d11,d12,d13}.

Two expansions to the sense of $SC_A^+(d_i)$ are:

$$\cup_{l=5}^{10} SC_A^+(d_l) \cup SC_A^{+'}(d_7) \cup SC_A^+(d_{13})$$

={d5,d6,d7,d8,d9,d10,d13};

$$\cup_{l=5}^{10} SC_A^+(d_l) \cup SC_A^{+''}(d_7) \cup SC_A^+(d_{13})$$

={d5,d6,d7,d8,d9,d10,d11,d13}.

Two expansions to the sense of $SC_A^-(d_i)$ are:

{d2,d4,d5,d6,d7,d8,d9,d10,d13};
{d5,d6,d7,d8,d9,d10,d13}.

Two expansions to the sense of $SC_A^0(d_i)$ are:

{d5,d6,d7,d8,d9,d10,d13};
{d5,d6, d7,d8,d9,d10,d12,d13}.

These reflect the diversity of expansions.

(3) Make $SU_A(d_i)$ and $SU_A^+(d_i)$, $SU_A^-(d_i)$, $SU_A^0(d_i)$ get expansions to the query. X={d5, d6,d7,d8,d9,d10,d13}. Expansions to the senses of $SU_A(d_i)$, $SU_A^+(d_i)$, $SU_A^-(d_i)$, $SU_A^0(d_i)$ are:

{d2,d4,d5,d6,d7,d8,d9,d10,d11,d12,d13},
{d5,d6,d7,d8,d9, d10,d11,d13},
{d2,d4,d5,d6,d7,d8,d9,d10,d13},
and {d5,d6,d7,d8,d9,d10,d12,d13} respectively.

2) Realize Elaborative queries

(1) Retrieve documents that certainly satisfy: '((a=18)&(d="s")) or ((b=10)&(d="m"))'. Owing to '((a=18)&(d="s"))or((b=10)& (d="m"))' , we can first retrieve the information system in Table 1(* is said to match any value). The initial retrieval result is X ={d5,d6,d7,d8,d9, d10,d13}. The answer to this query is $\underline{SIM}_A(X) = \underline{SIM}_A^+(X) \cap \underline{SIM}_A^-(X) \cap \underline{SIM}_A^0(X)$. Because

$\underline{SIM}_A^+(X)$ ={d5,d6,d7,d8,d9,d10,d11,d13},

$\underline{SIM}_A^-(X)$ ={d2,d4,d5,d6,d7,d8,d9,d10,d11,d12,d13},

$\underline{SIM}_A^0(X)$ ={d5,d6, d7,d8,d9,d10,d13},

thus, $\underline{SIM}_A(X)$ ={d5,d6,d7,d8,d9,d10}.

TABLE III CLASS $SC_A(d_i)$ AND ITS THREE DECOMPOSITIONS

D	SC_A	SC_A^+	SC_A^-	SC_A^0
d1	d1-d4	d1-d3	d1	d1,d4
d2	d1-d4	d2,d3	d2,d1,d4	d2
	d2,d4, d5,d13	d2,d5, d13	d2,d4	
d3	d1-d4	d3,d4	d1-d3	d3
	d3,d12	d3,d12	d3	
d4	d1-d4	d4,d2	d3,d4	d4,d1
	d2,d4, d5,d13	d2,d4, d5,d13	d4	d4
d5	d2,d4, d5,d13	d5	d5,d2,d4	d5,d13
	d5,d6,d13		d5	d5,d6, d13
d6	d5,d6,d13	d6	d6	d5,d6, d13
d7	d7-d10	d7,d10	d7	d7-d9
	d7,d11	d7,d11		d7
d8	d7-d10	d8,d10	d8	d7-d9
d9	d7-d10	d9	d9	d7-d10
d10	d7-d10	d10	d7,d8,d10	d9,d10
d11	d7,d11	d11	d7,d11	d11
d12	d12,d13	d12	d12	d12,d13
	d3,d12		d3,d12	d12
d13	d2,d4, d5,d13	d13	d2,d4, d13	d5,d13
	d5,d6,d13		d13	d5,d6, d13
	d12,d13			d12,d13

(2) Retrieve those documents that certainly satisfy: a little bit better than those that satisfy '((a=18) &(d="s")) or

((b=10)&(d=”m”))’. According to the inner condition, X={d5,d6,d7,d8,d9,d10,d13} is directly retrieved from Table 1. However, the query may be answered by applying the above decomposition based on dominance relation. In reality, the answer is $\underline{SIM}_A^+(X)$ ={d5,d6,d7,d8,d9, d10, d11,d13} in which d11 is a little bit better than the condition ‘(b=10)&(d=”m”)’ and there is no other documents a little bit better than ‘(a=18) &(d=”s”)’ from Table 1.

(3) Retrieve those documents that probably satisfy : ‘not worse but not better much than or just only the same as those satisfying ‘((a=18)&(b=27) &(c=18) &(d=”s”))’. According to the inner condition given by a Boolean expression, we can get X={d5,d6}. The answer to the entire query is to find the union of $\overline{SIM}_A^+(X)$ and $\overline{SIM}_A^0(X)$. i.e., $\overline{SIM}_A^+(X) \cup \overline{SIM}_A^0(X)$ ={d5,d6,d13}.

(4) A query is to retrieve all documents mutually similar in the corpus satisfying ‘((a=18)&(d=”s”)) or ((b=10)&(d=”m”))’. t means that we have to find documents to the sense of $SC_A(d_i)$. The initial document subset also is X={d5,d6,d7,d8,d9,d10,d13}. Then, the expansion of X to $SC_A(d_i)$ is:

{d5,d6,d7,d8,d9,d10,d13},
{d2,d4,d5,d6,d7, d8,d9,d10, d13},
{d5,d6,d7,d8,d9,d10,d12,d13},
{d2,d4,d5,d6,d7,d8,d9,d10,d12,d13},
{d2,d4,d5,d6,d7,d8,d9,d10,d11,d13},
or {d2,d4,d5,d6,d7,d8,d9, d10,d11,d12,d13}.

TABLE IV CLASS $SL_A(d_i)$ AND ITS THREE DECOMPOSITIONS

D	SL_A	SL_A^+	SL_A^-	SL_A^0
d1	d1-d4	d1-d3	d1	d1,d4
d2	d1-d5,d13	d2	d2,d4	d2
d3	d3	d3	d3	d3
d4	d2,d4	d4	d2,d4	d4
d5	d5,d13	d5	d5	d5,d13
d6	d6,d5,d13	d6	d6	d5,d6,d13
d7	d7	d7	d7	d7
d8	d7-d10	d8,d10	d8	d7,d8,d9
d9	d7-d10	d9	d9	d7-d10
d10	d7-d10	d10	d7,d8,d10	d9,d10
d11	d7,d11	d11	d11,d7	d11
d12	d12	d12	d12	d12
d13	d13	d13	d13	d13

If a query contains some special requirements such as a little better or a little worse than the condition, answers to the query must be done with a retrieval to the sense of $SC_A^+(d_i)$ or $SC_A^-(d_i)$. Queries using $SC_A^+(d_i)$, $SC_A^-(d_i)$,$SC_A^0(d_i)$ and $SL_A(d_i)$, $SL_A^+(d_i)$, $SL_A^-(d_i)$, $SL_A^0(d_i)$ are omitted here. Obviously, These expansions enlarge overcalls and may bring diverse responses.

VII. CONCLUSIONS

This paper introduces granules $SU_A(d_i)$, $SC_A(d_i)$ and $SL_A(d_i)$ whose constructing processes are from the view points of using maximal compatible classes as primitive granules. It extends original rough set model based on these granules. Three different kinds of upper and lower approximation decompositions are put forward so as to make document expansions flexible and to give more precise answers to elaborative queries. Through an example, the powers to finish complicated or elaborative queries are demonstrated. It provides us with an important theoretical principle to solve the problem how tolerance rough set model takes effects in the incomplete information system retrieval. It will be our further research targets to construct a prototype incorporating these new ideas with our existed results[4] and implement other expansions such as term expansions in information retrieval or other areas such as medical or biological text mining to realize knowledge acquisition because the approach proposed here is general and scalable.

REFERENCES

[1] C.C.Chan,“A Rough Set Approach to Attribute Generalization in Data Mining”,Information Sciences, 107, 1998, pp.169-176.

[2] C.Wu, X.B.Yang, “Information Granules in General and Complete Covering”, Proceedings of 2005 IEEE International Conference on Granular Computing, pp.675-678.

[3] G.Li,X.Zhang, “Decomposition of Rough Set Based on Similarity Relation”, J. of Computer Engineering and Applications, 2,2004,pp. 85-96,179.

[4] J.Stefanowski,A.Tsoukiàs, “Incomplete Information Tables and Rough Classification”, J. Computational Intelligence, Vol. 17, 3 ,2001,pp.545-566.

[5] K. Funakoshi, T. B. Ho, “Information Retrieval by Rough Tolerance Relation”, Proceedings of the 4th International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery, November 6-8, 1996, Tokyo, Japan, pp. 31-35.

[6] M.Kryszkiewicz, “Rough Set Approach to Incomplete Information Systems”, Information Sciences, Vol.112,1,1998, pp.39-49.

[7] R.W. Winiarski,A.Skowron, “Rough Set Methods in Feature Selection and Recognition”,Pattern Recognition Letters, 24,2003,pp.833~849.

[8] V.S.Ananthanarayana,M.N.Murty and D.K.Subramanian, “Tree Structure for Efficient Data Mining Using Rough Sets”, Pattern Recognition Letters, 24,2003,pp.851-862.

[9] W.L.Chen,J.X.Cheng and C.J.Zhang, “A Generalization to Rough Set Theory Based on Tolerance Relation”, J. computer eng ineering and applications, 16,2004,pp.26-28.

[10] Y.Li,“A Fuzzy-Rough Model for Concept Based Document Expansion”, RSCTC 2004,LNAI 3066,pp.699-707.

[11] Z.Pawlak, “Rough sets and intelligent data analysis”, Information Sciences. 147,2002,pp.1-12.