Development of an Information Retrieval Method Based on the Rough Set Theory

By Kaname FUNAKOSHI

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Masayuki Kimura

February 14, 1997

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Information Retrieval

There exist a large number of documents in the world and the number is enlarged day after day. But the archived documents cannot be used by readers if they are not classified and retrieved suitably [25]. It is a labor of librarians (called searchers) to provide documents appropriate to the user. In the electronized libraries, which are rapidly growing up today, automatic retrieval systems are much required. Consequently, the urgent need for high quality automatic information selections is also increasing. Instead of searchers, information retrieval system is expected to select and provide documents which are appropriate to the user from large scale databases.

From a huge database, even if a good retrieval systems with better selecting strategies tend to provide too many documents whereas sometimes the user wants to get a few documents with very high pertinence. In this case, the retrieval system weighted on the precision rather than the recall is desirable.

An information retrieval system consists of some parts of facilities. Figure 1.1 shows the simple graphical view of an information retrieval system by Belkin and Croft [1]. Each text to be retrieved in database is changed to the surrogate (for the simpleness) which represents the original text. And information problem of the user is changed to the query. Information retrieval system compares query to the surrogate and provide the texts which is led by the surrogates. This work focuses on the conceptual model of information retrieval which relates to the comparison of the query and the surrogate text (i.e., the index terms of the document). The conceptual model determines the way to

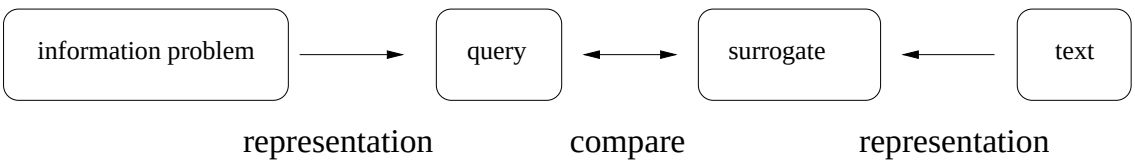

Figure 1.1: Retrieval technique situation (Belkin and Croft, 1989)

decide each document is whether relevant or irrelevant to the query.

Most of application information retrieval systems use Boolean operations (AND, OR, and NOT) as the conceptual model because it is easy to implement and it does not require long time to process a query [7]. However, these systems do not seem to be appropriate because Boolean operations do not always provide suitable documents. Their well known weaknesses are that they cannot deal with the related terms, cannot provide ranked outputs, cannot process term weights, and requires complex formulated queries.

Many non-Boolean information retrieval strategies have been investigated to overcome the limitations of Boolean operations. Among them, intelligent information retrievals carry out semantic calculations to select the documents. The information retrieval model by using the rough set theory investigated in this work is also in this direction.

1.2 Rough Sets and Rough Tolerance Relations

Rough set theory, a new mathematical tool to deal with vagueness and uncertainty introduced by Pawlak in early 1980s [20], has been successful in many real-life applications. This theory is an extension of the set theory in which each subset of a universe is described by a pair of ordinary sets called *lower* and *upper approximations*, determined by *equivalence relations* between two elements in the universe. The idea of using rough sets in information retrieval has been addressed early in several works, e.g., Raghavan and Sharma [24] and Srinivasan [33], [34]. However, the requirement of reflexive, symmetric and transitive properties in equivalence relations, which is suitable in many application domains, is too strict in the field of information retrieval where the transitive property is not always satisfied. Therefore these early works seem not appropriate though the basic notions are very powerful.

Recently some approaches employing tolerance relations which do not require transitive property to generalize the model of the rough set theory have been investigated, e.g., [31], [39]. By using the new methods on the rough sets as the conceptual model, information retrieval systems will be more satisfactory for users.

1.3 Organization of the Thesis

This thesis presents a work on a conceptual model of information retrieval based on rough set theory using tolerance relations instead of equivalence relations. The main contributions of this work are the formulation of this conceptual model and the determination of a matching algorithm based on the rough tolerance inclusions. The method has been implemented and tested with a database in order to prove its advantages and application potential. This is the case study with the database of articles in the Journal of the Japanese Society for Artificial Intelligence [10], [12].

This thesis consists of seven chapters. Chapter 2 describes an information retrieval system and its formulation. It at first reviews works on conceptual models of information retrieval, and then makes a definition of an information retrieval system. Chapter 3 introduces the

rough set theory as the technical basis for the information retrieval. At first the traditional rough sets with equivalence relations are introduced and the common features of rough sets are pointed out. Relations between two subsets used in early works and extended in this work are also presented. Then the generalized rough sets with tolerance relations and the extensions of the common features as relations used in this work will be described. Chapter 4 introduces the proposed method of information retrieval by using the rough tolerance relations which is the objective of this thesis. Chapter 5 presents the architecture and implementation of the system. The characteristics of the database used in the system and some assumptions to construct the system are emphasized. Chapter 6 reports the experimental results and the evaluation of the system constructed in previous chapter. The evaluation of the method is also discussed. Chapter 7 gives the conclusion of the method and the system. Some unreached points are enumerated.

Chapter 2

Information Retrieval Systems

2.1 Introduction to Information Retrieval Systems

An information retrieval system is an information system which requires queries as the input and provides sets of bibliographical data as the output [30]. Each bibliographical data contains an identifier of a document like the author names, title, journal name, volume, number, pages, publisher, publish year, etc. While the relation between the bibliographical data and the real document is a bijection, each bibliographical data identifies and represents a document. So the bibliographical data can be called *documents* in this work.

An information retrieval system consists of many parts such as file structure, query operations, term operations, document operations, hardware, and conceptual model [8]. This work focuses on a conceptual model which determines the strategy to select relevant documents (represented as a set of index terms) to the user query. The query and the index terms are assumed to be settled properly.

2.2 Conventional Conceptual Models of Information Retrieval

Most traditional conceptual models of information retrieval are Boolean models which uses Boolean operations to compare the query and the documents. Although some information retrieval systems based on intelligent models are available through the computer

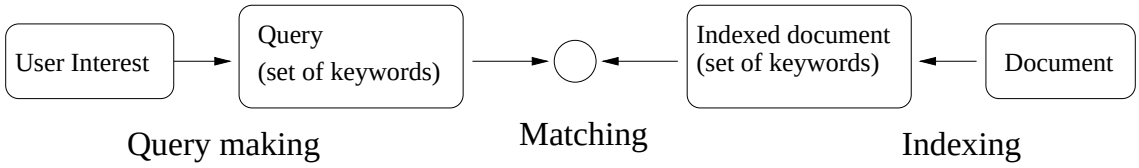

Figure 2.1: Overview of an information retrieval system

Retrieval techniques	Ranked output	Related terms	Term weight	Simple query	Cost
Boolean	not available	no	not available	no	good
Vector Space	sensitive	yes	available	in case	average
Fuzzy Set	not sensitive	yes	available	no	average
Probability	sensitive	yes	available	in case	bad
Spreading Activation	sensitive	yes	available	in case	bad
Rough Sets	not sensitive	yes	may be	yes	good

Table 2.1: Information retrieval techniques

network, most of conventional information retrieval systems use Boolean operations as the conceptual models yet. The main advantage of Boolean operations lies in their simplicity, but it is well known that they do not always provide good responses to the user's interest. While Boolean operations on sets which are means of expressing exactly queries in information retrieval systems [37] have been criticized, improving their retrieval effectiveness has been difficult.

There have been attempts to improve the information retrieval quality by doing inexact match with different techniques such as probabilistic models, vector space models, intelligent retrieval models, among others. Intelligent matching strategies for information retrieval often use the concept analysis requiring semantic calculations at different levels [13].

Several intelligent information retrieval strategies have been investigated. Fuzzy set models and connectionist models are traditional intelligent retrieval techniques and they have been studied for many years. Some other models, e.g., genetic algorithm models [4] and case-based reasoning models [32], [28] have been studied in recent years. The rough sets model provides also an intelligent conceptual model for information retrieval.

Table 2.1 resumes the advantages and disadvantages of commonly used retrieval techniques reviewed or evaluated by Wong and Raghavan [38], Bookstein [3], Belkin and Croft [1], Doschzkocs et al. [5], Koll and Srinivasan [14], Taniguchi [36] and our evaluation on those of the rough set model. We believe that rough set theory, with its soft computing power, will contribute a good solution to the field of information retrieval.

2.3 Formulation of Information Retrieval Systems

In connection to the presentation of rough tolerance retrieval systems, an information retrieval system is formulated as follows. This retrieval system consists of documents, queries and selecting strategy represented as a function. In this work the information retrieval system is considered to be a tag-based retrieval system which provides meta-information such as terms indexed to the specific documents [6]. So it is suitable that both documents and the queries are regarded as a set of keywords. From the above notice,

the information retrieval system consists of a set of keywords, a set of documents, a set of queries and a retrieval function. The tag-based information retrieval systems will be formulated generally.

Information retrieval systems can be formulated as a quadruple

$$\mathcal{S} \triangleq (\mathcal{T}, \mathcal{D}, \mathcal{Q}, \alpha) \tag{2.1}$$

where

$$\mathcal{T} \triangleq \{t_1, t_2, \ldots, t_M\} \tag{2.2}$$

is a set of index terms (i.e., keywords);

$$\mathcal{D} \triangleq \{d_1, d_2, \ldots, d_N\} \tag{2.3}$$

is a set of documents where $d_j \subseteq \mathcal{T}$; $\mathcal{Q}$ is a set of queries with each query $Q \subseteq \mathcal{T}$; and $\alpha : \mathcal{Q} \times \mathcal{D} \rightarrow \mathbb{R}^+$ is a retrieval function between a query and a document. In a more general form a document d_j can be denoted as a set of index term-weight pairs

$$d_j \triangleq (t_1, w_{t_1};\ t_2, w_{t_2};\ \ldots;\ t_n, w_{t_n}) \tag{2.4}$$

where $t_i \in \mathcal{T}$ and $w_{t_i} \in [0, 1]$ reflects the relative importance of term t_i in d_j. A query Q can also be denoted as a set of index term-weight pairs

$$Q \triangleq (q_1, w_{q_1};\ q_2, w_{q_2};\ \ldots;\ q_m, w_{qm}) \tag{2.5}$$

where $q_i \in \mathcal{T}$ and $w_{q_i} \in [0, 1]$. This work does not deal with the weighted terms yet but can be extended to have the term weights by using this general form. The information retrieval task is to give the answer as a set

$$\mathcal{A} \triangleq \{d_{a_1}, d_{a_2}, \ldots, d_{an}\} \subseteq \mathcal{D} \tag{2.6}$$

to the query Q, with decreasing order of $\alpha(Q, d_{a_i})$.

Chapter 3

Rough Sets

3.1 Basics of the Rough Set Theory

3.1.1 Basis Notions of Rough Sets

The rough set theory was introduced by Pawlak [20] in early 1980s as a new mathematical tool to deal with vagueness and uncertainty. It is applied in many branches of artificial intelligence, especially to machine learning, knowledge acquisition, decision analysis, knowledge discovery in database, expert systems, decision support systems, inductive reasoning, and pattern recognition [22]. The starting point of the rough set theory is an indiscernibility relation. If two elements have the same value for an attribute, they are regarded as *indiscernible* at the attribute. The primary notions of the theory are *approximation space* and *lower* and *upper approximations* of a set [17].

The universe U is a set of elements where each element $x \in U$ has a value $f_a(x)$ for each attribute a in a set of attributes F where $f_a : U \rightarrow V_a$ is called an *information function*, V_a is the set of values of a called *domain* of the attribute a [21]. An *equivalence relation* R on U can be defined for each set of attributes $B \subseteq F$ as follows

$$R \triangleq \{\ (x, y)\ \in U \times U \mid f_a(x) = f_a(y),\quad \forall\, a \in B\} \tag{3.1}$$

Denote by $U/R \subset \mathcal{P}(U)$ the approximation space of U regarding the equivalence relation R. The approximation space U/R contains a set of categories each is a subset of indiscernible elements of U, called *equivalence classes*. As the equivalence relation R satisfies reflexive, symmetric and transitive properties, equivalence classes of U regarding R are disjoint categories.

Lower and upper approximations are subsets of the universe. For every subset $X \subseteq U$ the lower and upper approximations are assigned. The lower approximation of X denoted by $\mathcal{L}_R(X)$ is a union of the categories which are included in X and the upper approximation of X denoted by $\mathcal{U}_R(X)$ is the union of the categories whose intersections with X are not empty. These are defined as follows

$$\mathcal{L}_R(X) \triangleq \bigcup\{Q \in U/R \mid Q \subseteq X\} \tag{3.2}$$

$$\mathcal{U}_R(X) \triangleq \bigcup\{Q \in U/R \mid Q \cap X \neq \varnothing\} \tag{3.3}$$

Element	*Attribute*
e_1	a_1
e_2	a_1
e_3	a_1
e_4	a_2
e_5	a_2
e_6	a_3
e_7	a_3
e_8	a_4
e_9	a_4
e_{10}	a_4
e_{11}	a_5

Table 3.1: Sample table of elements and their attributes

Before applying rough set theory to the information retrieval, a simplified example of rough set for the purpose of information retrieval is given. Table 3.1 shows a sample universe U with 11 elements. Each element is described by only one attribute a with five values a_1, a_2, a_3, a_4, a_5. Figure 3.1 shows a graphical image of the universe where elements are indicated according to their attribute values.

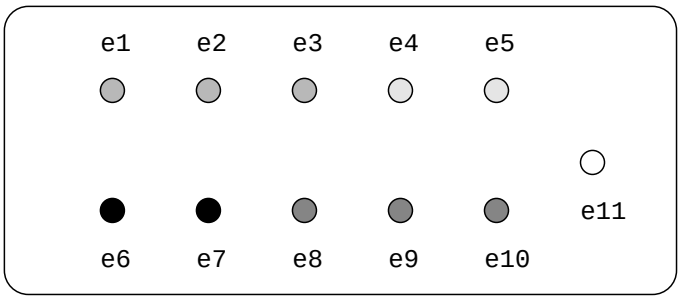

Figure 3.1: Sample universe

The universe is divided into five categories (the number of categories depends on values of the attribute). Each element $e_i \in U$ is classified into a category according to the value for the attribute $a(e_i)$ (Figure 3.2). Each category is an equivalence class and is not distinguishable. All elements in the same equivalence class have an equivalence relation. The important idea of the rough set theory is that the elements which have the same value for the attributes are regarded as equivalent. This separated universe is the approximation space.

For any subset $X \subseteq U$, the lower and upper approximations can be assigned. For example, given a subset $X = \{e_1, e_2, e_6, e_7, e_8\}$, the lower approximation is assigned as $\mathcal{L}_R(X) = \{e_6, e_7\}$ and the upper approximation is assigned as $\mathcal{U}_R(X) = \{e_1, e_2, e_3, e_6, e_7, e_8, e_9, e_{10}\}$. Figure 3.3 shows graphically the lower and upper approximations of X.

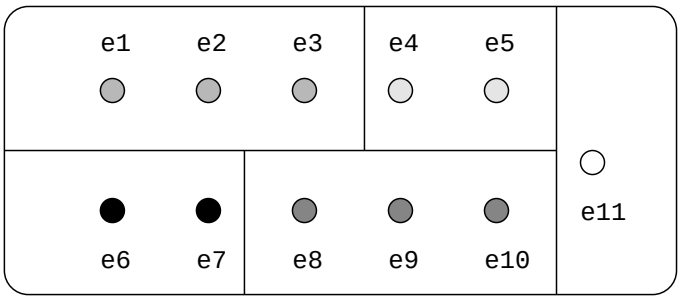

Figure 3.2: The universe divided into categories

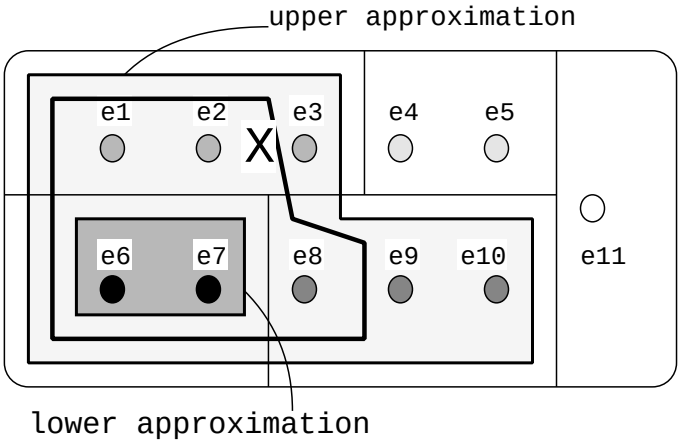

Figure 3.3: Lower / Upper Approximations of X

3.1.2 Relation Between Equivalence Classes

There are several kinds of inclusion relationships between two subsets in the universe according to their equivalence classes for equivalence relation R. Three of these relationships are used in this work, namely *rough equality*, *rough inclusion* and *rough overlap*. Rough equality and rough inclusion are defined in [20]. There are three kinds of rough equalities called *rough bottom equality*, *rough top equality* and *rough equality*. For any two subsets $X, Y \subseteq U$, if the lower approximations of X and Y are equal and not empty, they are called *roughly bottom equal*, and denoted by $X \eqsim Y$. If the upper approximations of X and Y are the same, they are called *roughly top equal* and denoted by $X \simeq Y$. And if X and Y are both roughly bottom and top equal, they are called *roughly equal* and denoted by $X \approx Y$. They are defined as follows

$$X \eqsim Y \triangleq X, Y \subseteq U \wedge \mathcal{L}_R(X) = \mathcal{L}_R(Y) \tag{3.4}$$

$$X \simeq Y \triangleq X, Y \subseteq U \wedge \mathcal{U}_R(X) = \mathcal{U}_R(Y) \tag{3.5}$$

$$X \approx Y \triangleq X \simeq Y \wedge X \eqsim Y \tag{3.6}$$

Similarly, there are three rough inclusions called *rough bottom inclusion*, *rough top inclusion* and *rough inclusion*. For two subsets $X, Y \in U$ if the lower approximation of X is

a subset of the lower approximation of Y, X is called *roughly bottom included* in Y. If the upper approximation of X is a subset of the upper approximation of Y, X is called *roughly top included* in Y. If X is both roughly bottom and top included in Y, X is called *roughly included* in Y. They are denoted by following

$$X \underset{\sim}{\subseteq} Y \quad \triangleq \quad X, Y \subseteq U \quad \wedge \quad \mathcal{L}_R(X) \subseteq \mathcal{L}_R(Y) \tag{3.7}$$

$$X \tilde{\subseteq} Y \quad \triangleq \quad X, Y \subseteq U \quad \wedge \quad \mathcal{U}_R(X) \subseteq \mathcal{U}_R(Y) \tag{3.8}$$

$$X \underset{\sim}{\tilde{\subseteq}} Y \quad \triangleq \quad X \underset{\sim}{\subseteq} Y \ \wedge \ X \tilde{\subseteq} Y \tag{3.9}$$

And there are two rough overlaps called *roughly bottom overlap* and *roughly top overlap*. For two subsets $X, Y \in U$, if the intersection of the lower approximations of X and Y is not empty, that means $\mathcal{L}_R(X) \cap \mathcal{L}_R(Y) \neq \varnothing$, then X and Y are called *roughly bottom overlapping*. If the intersection of the upper approximations of X and Y are not empty, that means $\mathcal{U}_R(X) \cap \mathcal{U}_R(Y) \neq \varnothing$, then X and Y are called *roughly top overlapping* (no special notations for these overlaps).

3.1.3 Applications of Rough Sets to Information Retrieval

There are several works of information retrieval by using the traditional rough set theory by Raghavan and Sharma [24] and Srinivasan [33], [34]. These studies use the previous relationships between the user query and the documents in the database. In the work of Srinivasan [33], retrieval is carried out by 13 levels in order rough equalities, rough inclusions, and rough overlaps.

3.2 Generalized Approximation Spaces

The traditional rough set theory using equivalence relations requires indistinguishable categories because the equivalence relations require three properties which are reflexive, symmetric and transitive. Relation $R \subseteq U \times U$ is equivalent if it satisfies following properties, for all $x, y, z \in U$

- reflexive: xRx
- symmetric: $xRy \rightarrow yRx$
- transitive: $xRy \wedge yRz \rightarrow xRz$

These properties are often hold in many application fields but all the properties do not always hold in certain application domains. Especially in the fields of linguistic or information retrieval, the transitive property is too strict and rough sets using the equivalence relation cannot be well applied. Intelligent information retrieval methods rely mainly on the conceptual analysis and they require considering relationships between the terms. Sometimes each term can be replaced by another term with similar concept, but cannot be always replaced by the same term because there is no such thing as a *true* synonym

which have exactly the same meaning. Only case that the term can be always replaced by another term which is indiscernible to the original term is the synonyms such as a relation between "artificial intelligence" and "AI". A previous work [34] uses rough equivalence relations as to deal with these synonym relations. Even if the stemming relations [9] such as "user" and "users" cannot always be exchanged each other.

To resolve this difficulty, generalized approximation spaces without the condition of transitive property are introduced. Relations which require only the reflexive and symmetric properties are named *tolerance relations* (or compatible relation) and have been introduced in [31], [39], [23]. The generalized approximation space with such tolerance relations is called *tolerance spaces*. Figure 3.4 shows the explanation of a tolerance space by Roget's thesaurus [29] [1].

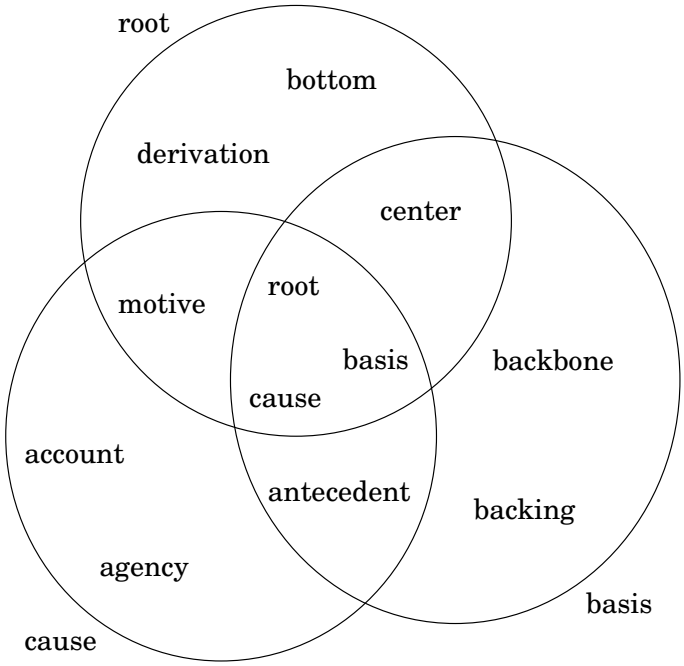

Figure 3.4: Tolerance Space

3.2.1 Definition of Generalized Approximation Spaces

In the work of Skowron and Stepaniuk [31], a tolerance space was defined as a quadruple $\mathcal{R} = (U, I, \nu, P)$, where U is a non-empty set of objects, $I : U \to \mathcal{P}(U)$ is an uncertainty function, $\nu : \mathcal{P}(U) \times \mathcal{P}(U) \to [0, 1]$ is a vague inclusion and $P : I(U) \to \{0, 1\}$ is a structurality function.

The uncertainty function I on U is any function satisfying the condition $x \in I(x)$ and $y \in I(x)$ iff $x \in I(y)$ for $\forall\, x, y \in U$. This function corresponds to a relation $\mathbb{I} \subseteq U \times U$

[1] This example is picked up carefully because sometimes this thesaurus does not even satisfy the symmetric property. This is a reason we do not use relations led form thesauri in this work.

understood as $x \mathbb{I} y$ iff $y \in I(x)$. $\mathbb{I}$ is a tolerance relation as it satisfies the properties of reflexivity and symmetry.

The vague inclusion $\nu : \mathcal{P}(U) \times \mathcal{P}(U) \rightarrow [0,1]$ defines the value of inclusion between two sets $X, Y \subseteq U$, according to the vagueness $\beta \quad (0 \leq \beta < 0.5)$. Denote by $t = \mid X \cap Y \mid / \mid X \mid$, the value of vague inclusion $\nu_\beta(X, Y)$ can be expressed as

$$\nu_\beta(X, Y) = \begin{cases} 0, & \text{if } \ 0 \leq t \leq \beta \\ f(t), & \text{if } \ \beta \leq t \leq 1 - \beta \\ 1, & \text{if } \ 1 - \beta \leq t \leq 1 \end{cases} \tag{3.10}$$

where $f(t)$ is any monotonous function in $\beta \leq t \leq 1 - \beta$. At last, $P : I(U) \rightarrow \{0, 1\}$ classifies $I(x)$ for each $x \in U$ into two classes – structural subsets $(P(I(x)) = 1)$ and non-structural subsets $(P(I(x)) = 0)$.

With the tolerance space $\mathcal{R}$, the lower approximation $\mathcal{L}$ and the upper approximation $\mathcal{U}$ for any $X \subseteq U$ are defined as

$$\mathcal{L}(\mathcal{R}, X) = \{x \in U \mid P(I(x)) = 1 \ \wedge \ \nu_\beta(I(x), X) = 1\} \tag{3.11}$$

$$\mathcal{U}(\mathcal{R}, X) = \{x \in U \mid P(I(x)) = 1 \ \wedge \ \nu_\beta(I(x), X) > 0\} \tag{3.12}$$

The basic problem of using tolerance spaces in any application is how to determine suitably I, ν and P.

3.2.2 Relation Between Tolerance Classes

Relationships between two tolerance classes are re-defined similarly those between equivalence classes defined in subsection 3.1.2. There are three rough relations between every two subsets $X, Y \in U$ which are *rough tolerance equality*, *rough tolerance inclusion* and *rough tolerance overlap*. There are three rough tolerance equalities called *rough bottom tolerance quality*, *rough top tolerance equality* and *rough tolerance equality* [18]. If the lower approximations of X and Y are called *roughly bottom tolerance equal*, and denoted by $X \eqsim Y$. If the upper approximations of X and Y are the same, they are called *roughly top tolerance equal* and denoted by $X \simeq Y$. And if X and Y are both roughly bottom and top tolerance equal, they are called *roughly tolerance equal*, denoted by $X \approx Y$. These relations can be formulated with our notations as follows

$$X \eqsim Y \ \triangleq \ X, Y \subseteq U \ \wedge \ \mathcal{L}(\mathcal{R}, X) = \mathcal{L}(\mathcal{R}, Y) \tag{3.13}$$

$$X \simeq Y \ \triangleq \ X, Y \subseteq U \ \wedge \ \mathcal{U}(\mathcal{R}, X) = \mathcal{U}(\mathcal{R}, Y) \tag{3.14}$$

$$X \approx Y \ \triangleq \ X \simeq Y \ \wedge \ X \eqsim Y \tag{3.15}$$

Similarly, there are three rough tolerance inclusions called *rough bottom tolerance inclusion*, *rough top tolerance inclusion* and *rough tolerance inclusion*. If the lower approximation of X is a subset of the lower approximation of Y, X is called *roughly bottom tolerance included* in Y. If the upper approximation of X is a subset of the upper approximation of Y, X is called *roughly top tolerance included* in Y. If X is both roughly bottom and top tolerance included in Y, X is called *roughly tolerance included* in Y. They are denoted

as follows

$$X \underset{\sim}{\subseteq} Y \quad \triangleq \quad X, Y \subseteq U \quad \wedge \quad \mathcal{L}(\mathcal{R}, X) \subseteq \mathcal{L}(\mathcal{R}, Y) \tag{3.16}$$

$$X \tilde{\subset} Y \quad \triangleq \quad X, Y \subseteq U \quad \wedge \quad \mathcal{U}(\mathcal{R}, X) \subseteq \mathcal{U}(\mathcal{R}, Y) \tag{3.17}$$

$$X \underset{\sim}{\tilde{\subseteq}} Y \quad \triangleq \quad X \underset{\sim}{\subseteq} Y \ \wedge \ X \tilde{\subset} Y \tag{3.18}$$

And there are two rough tolerance overlaps called *roughly bottom tolerance overlap* and *roughly top tolerance overlap*. If the intersection of the lower approximations of X and Y is not empty, that is $\mathcal{L}(\mathcal{R}, X) \cap \mathcal{L}(\mathcal{R}, Y) \neq \varnothing$, then X and Y are called *roughly bottom tolerance overlapping* and if the intersection of the upper approximations of X and Y are not empty, that is $\mathcal{U}(\mathcal{R}, X) \cap \mathcal{U}(\mathcal{R}, Y) \neq \varnothing$, then X and Y are called *roughly top tolerance overlapping*. There are no notations for rough tolerance overlaps.

Chapter 4

A Tolerance Relation Based Method for Information Retrieval

4.1 Determination of the Rough Tolerance Space

The essence of the method is how to determine suitably I, ν and P defined in subsection 3.2.1 for the information retrieval problem. First of all, to determine a tolerance space $\mathcal{R}$, the universe U is chosen as the set $\mathcal{T}$ of all terms in the database $\mathcal{D}$

$$U \triangleq \{t_1, t_2, \ldots, t_M\} = \mathcal{T} \tag{4.1}$$

The key notion used in the method is the *co-occurrency of terms* in all documents from the database. Some information retrieval models based on probabilities use the co-occurrency of the terms to gather the keywords and select the documents [26]. The co-occurrency is useful for our study because it does not only satisfies both reflexive and symmetric properties, but also provides a way for creating semantic network in the database. Co-occurrency is defined as follows. Denote by $C : \mathcal{T} \times \mathcal{T} \rightarrow \mathcal{P}(\mathcal{D})$ the function that determines the set of all documents in which a pair of term co-occurs. It means that for any two terms $t_i, t_j \in \mathcal{T}$

$$C(t_i, t_j) \triangleq \{d \in \mathcal{D} \mid t_i \in d \ \wedge \ t_j \in d\} \tag{4.2}$$

Denote by $c : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{N}$ the function which determines the frequency of co-occurrence of two terms in the database. it means that for any two terms $t_i, t_j \in \mathcal{T}$

$$c(t_i, t_j) \triangleq \ \mid C(t_i, t_j) \mid \tag{4.3}$$

The notation $c(x, x)$ for any $x \in \mathcal{T}$ denotes the frequency of the term x in the database.

The uncertainty function I is defined depending on a threshold θ as follows

$$I_\theta(t_i) \triangleq \{t_j \mid c(t_i, t_j) \geq \theta\} \cup \{t_i\} \tag{4.4}$$

It is clear that the relation $c(t_i, t_j)$ defined above is both reflexive and symmetric, so the function I satisfies the requirements of an uncertainty function on $\mathcal{R}$ defined in subsection

3.2.1. This function corresponds to a tolerance relation $\mathbb{I} \subseteq U \times U$ defined by $t_i \mathbb{I} t_j$ iff $t_j \in I_\theta(t_i)$. $I_\theta(t_i)$ is the tolerance class of term t_i. Also denote by $c(t_i, t_i)$ the number of occurrences for term t_i in the database $\mathcal{D}$. The vague inclusion function ν is defined as

$$\nu(X, Y) \triangleq \frac{| X \cap Y |}{| X |} \tag{4.5}$$

The membership function for $t_i \in \mathcal{T}, X \subseteq U$ is defined by

$$\mu(t_i, X) \triangleq \nu(I_\theta(t_i), X) = \frac{| I_\theta(t_i) \cap X |}{| I_\theta(t_i) |} \tag{4.6}$$

All the tolerance classes of terms in the database are classified into structural subsets. So for any $t_i \in \mathcal{T}$, $P(I_\theta(t_i)) = 1$. With all above definitions, we have the tolerance space $\mathcal{R} = (U, I, \nu, P)$. In this tolerance space $\mathcal{R}$, the lower tolerance approximation $\mathcal{L}$ and the upper tolerance approximation $\mathcal{U}$ for any subset $X \subseteq \mathcal{T}$ are defined by

$$\mathcal{L}(\mathcal{R}, X) \triangleq \{ t_i \in \mathcal{T} \mid \nu(I_\theta(t_i), X) = 1\} \tag{4.7}$$

$$\mathcal{U}(\mathcal{R}, X) \triangleq \{ t_i \in \mathcal{T} \mid \nu(I_\theta(t_i), X) > 0\} \tag{4.8}$$

4.2 Rough Tolerance Matching of Documents

The matching between the user query and documents can be now carried out by checking different levels of rough inclusions (involving equality and overlap) between their tolerance lower and upper approximations. The rough inclusions between two sets for tolerance relations described in subsection 3.2.2 are used for the matching. There are totally 12 levels of inclusions between two sets which can appear while matching the set of terms in the user query Q to the set of terms in each documents d_j. The retrieval is executed by 12 levels of conditions with inclusions as follows.

(1) Definability: This simple level is certainly the best match, but occurs rarely in real-world problems

$$Q = d_j \qquad \text{[1-1]}$$

(2) Rough equalities:

$$\mathcal{L}(\mathcal{R}, Q) \neq \varnothing \quad \wedge \quad Q \approx d_j \qquad \text{[2-1]}$$

$$\mathcal{L}(\mathcal{R}, Q) \neq \varnothing \quad \wedge \quad Q \approx\!\!\!- d_j \qquad \text{[2-2]}$$

$$Q \simeq d_j \qquad \text{[2-3]}$$

(3) Rough inclusions:

$$\mathcal{L}(\mathcal{R}, Q) \neq \varnothing \quad \wedge \quad Q \underset{\sim}{\overset{\sim}{\subseteq}} d_j \qquad \text{[3-1]}$$

$$\mathcal{L}(\mathcal{R}, Q) \neq \varnothing \quad \wedge \quad Q \underset{\sim}{\subseteq} d_j \qquad \text{[3-2]}$$

$$Q \overset{\sim}{\subset} d_j \qquad \text{[3-3]}$$

(4) Rough inclusions (opposite of 3): Other situations may occur as in *(3)* but the role of Q and d_j are inversed

$$\mathcal{L}(\mathcal{R}, d_j) \neq \varnothing \quad \wedge \quad d_j \underset{\sim}{\overset{\sim}{\subseteq}} Q \qquad \text{[4-1]}$$

$$\mathcal{L}(\mathcal{R}, d_j) \neq \varnothing \quad \wedge \quad d_j \underset{\sim}{\subseteq} Q \tag{4-2}$$

$$d_j \overset{\sim}{\subset} Q \tag{4-3}$$

(5) Rough overlaps: Finally, it may happen that the tolerance lower and upper approximations of Q and d_j are overlapping

$$\mathcal{L}(\mathcal{R}, Q) \cap \mathcal{L}(\mathcal{R}, d_j) \neq \varnothing, \tag{5-1}$$

$$\mathcal{U}(\mathcal{R}, Q) \cap \mathcal{U}(\mathcal{R}, d_j) \neq \varnothing \tag{5-2}$$

The reasons of the conditions $\mathcal{L}(\mathcal{R}, Q) \neq \varnothing$ and $\mathcal{L}(\mathcal{R}, d_j) \neq \varnothing$ in [2-1], [2-2], [3-1], [3-2], [4-1], [4-2] are that any empty sets are equally and any set include the empty set [15], i.e.,

$$\varnothing \quad = \quad \varnothing \tag{4.9}$$

$$\forall X \subseteq U \quad \rightarrow \quad \varnothing \subseteq X \tag{4.10}$$

Without these conditions, for a query with empty lower approximation the system always retrieves all documents in the database in [2-2] or [4-2] and documents with empty lower approximation are always retrieved in [2-2] or [3-2]. In the levels [2-1] and [2-2], the condition $\mathcal{L}(\mathcal{R}, d_j) \neq \varnothing$ can be omitted. These levels are completed as follows

$$\mathcal{L}(\mathcal{R}, d_j) \neq \varnothing \quad \wedge \quad \mathcal{L}(\mathcal{R}, Q) \neq \varnothing \quad \wedge \quad Q \approx d_j \tag{2-1a}$$

$$\mathcal{L}(\mathcal{R}, d_j) \neq \varnothing \quad \wedge \quad \mathcal{L}(\mathcal{R}, Q) \neq \varnothing \quad \wedge \quad Q \eqsim d_j \tag{2-2a}$$

Denote by A_{11}, A_{21},..., A_{52} the sets of all documents satisfying conditions [1-1], [2-1],..., [5-2], respectively, when matching them against Q. It means that

$$A_{kl} \triangleq \{d_j \in \mathcal{D} \mid d_j \neq \varnothing \quad \wedge \quad d_j \text{ and } Q \text{ satisfy condition } [k\text{-}l]\} \tag{4.11}$$

The relevance degree to Q of documents in sets A_{11}, A_{21}, A_{22}, A_{23}, A_{31}, A_{32}, A_{33}, A_{41}, A_{42}, A_{43}, A_{51}, A_{52} is decreasing in this order of these sets, called *relevance rank*. This rank shows that A_{11} is the set of the most relevant documents to Q, then A_{21} and so on. Essentially, our answer to the user query Q is a sequence of these ordered sets in matching all documents $d_j \in \mathcal{D}$ with Q. The corresponding algorithm is formulated as follows

Algorithm *Matching*$(Q, \mathcal{D})$
begin
$A_{11} \leftarrow \varnothing, A_{21} \leftarrow \varnothing, \ldots, A_{52} \leftarrow \varnothing$
if $Q \neq \varnothing$ **then**
begin
for $j = 1$ **to** $\mid \mathcal{D} \mid$ **do**
begin
if $d_j \neq \varnothing$ **then**
begin
if $Q = d_j$ **then** $A_{11} \leftarrow A_{11} \cup \{d_j\}$ `[1-1]Definitably`
if $\mathcal{L}(\mathcal{R}, Q) \neq \varnothing$ **and** $Q \approx d_j$ **then**
$A_{21} \leftarrow A_{21} \cup \{d_j\}$ `[2-1]rough equality`
if $\mathcal{L}(\mathcal{R}, Q) \neq \varnothing$ **and** $Q \eqsim d_j$ **then**
$A_{22} \leftarrow A_{22} \cup \{d_j\}$ `[2-2]rough bottom equality`
if $Q \simeq d_j$**then**
$A_{23} \leftarrow A_{23} \cup \{d_j\}$ `[2-3]rough top equality`
...
`Here rough inclusion matching is done`
...
if $\mathcal{L}(\mathcal{R}, Q) \cap \mathcal{L}(\mathcal{R}, d_j) \neq \varnothing$ **then**
$A_{51} \leftarrow A_{51} \cup \{d_j\}$ `[5-1]rough bottom overlap`
if $\mathcal{U}(\mathcal{R}, Q) \cap \mathcal{U}(\mathcal{R}, d_j) \neq \varnothing$ **then**
$A_{52} \leftarrow A_{52} \cup \{d_j\}$ `[5-2]rough top overlap`
end
end
end
end

4.3 Secondary Ranking on Rough Overlaps

The matching algorithm in section 4.2 does not reflect the degree of inclusions and overlaps and it results a discrete ranking of answer documents like fuzzy set models which differs from other ranking methods [3]. This ranking has a disadvantage in the levels of rough overlaps [5-1] and [5-2], which sometimes select many documents in the same answer sets A_{51} and A_{52} with different degrees of relevance. An elegant way to overcome this limitation is to introduce a term weight mechanism into the method. But it is difficult to obtain weights of the terms suitably as mentioned in a fuzzy sets model [16]. As another way to overcome this limitation, a secondary ranking of documents is introduced to the method specially in these two levels ([5-1] and [5-2]) by dividing them into subgroups each contains documents with the same degree of relevance.

The secondary ranking is obtained by applying the association measure of automatic classification of the terms [27]. The simplest association measure is $\mid Q \cap d_j \mid$ called simple matching coefficient and this can be obtained easily. Similarly with this measure, the vague inclusion function ν defined in (4.5) is used for the secondary ranking measure.

By this, each document d_j is assigned to one of $\mid Q \mid +1$ subgroups depending on the value of

$$\nu(Q, d_j) = \frac{\mid Q \cap d_j \mid}{\mid Q \mid} \tag{4.12}$$

It is clear that each of these $\mid Q \mid +1$ subgroups contains documents with the same degree of relevance. Relevance degrees of these subgroups are ranked according to the number of common keywords between the query and documents in each subgroup.

Chapter 5

Implementation and Case-study

5.1 Architecture of the System

The method determined in chapter 4 has been implemented in a system and tested with a real database. Figure 5.1 shows the overview of the system.

In this architecture, the system functions in two following phases. The first phase aims at calculating the tolerance relations of all terms and approximations of all documents (shown by the dashed arrows). The second phase aims at retrieving the documents for the user query (shown by the solid arrows). In the first phase, the tolerance space of documents is determined after creating or updating the database ($\mathcal{D}$ and $\mathcal{T}$). The system counts co-occurrencies of all terms from the database and calculates the tolerance classes of each term regarding different values of θ (from 1 to 4). Next, the system determines the upper and lower approximations for each document. In the second phase of retrieval, when a query is encountered, the system firstly calculates its upper and lower approximations based on the tolerance relations obtained in the first phase for each term in the query. Then the system determines twelve levels of matching ([1-1] to [5-2]) by using the rough tolerance inclusions as described in the algorithm, and results the answer as the sequence of non-empty sets among $A_{11}, ..., A_{52}$. These phases have been implemented in C.

To implement the method, some conditions are assumed for the simplicity. The keywords which are added to documents are assumed to be appropriate to characterizing the documents, i.e., this system requires the keywords suitable to the documents.

5.2 Data Source

The method is illustrated by a case study of retrieving relevant documents in a concrete database for the user query. The database is constructed by a real world data source of the Journal of Japanese Society for Artificial Intelligence (JSAI) after its first 10 years of publication (1986-1995). In this source, each document has fields of author name, title (both in Japanese and English), publish year, volume, number and pages. Each document is regarded as a pair of a name and a set of keywords which are added by the authors in

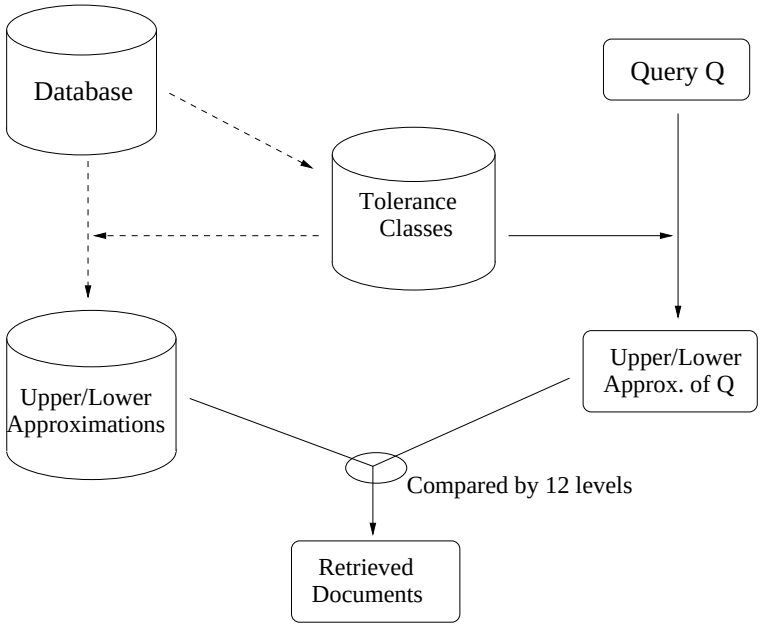

Figure 5.1: Architecture of the system

advance. Several documents with no keywords only have a name and its keyword is an emptyset (they are never retrieved in this implementation).

This database consists of 802 documents as partially described in Table 5.1, in which 725 documents are with keywords. Originally, there are 1883 keywords in the database. Several keywords, which are synonym or with the same stems, are manually set to one keyword. For example, terms "neural network", "neural networks", "neural-net" and "neuralnetwork" are set to "neural network". Using simple stemming operations, it remains 1813 keywords in the database (number of tolerance classes). As this operation is done by hand, some equivalent keywords remain in the database yet.

The indexing keywords of the journal are specified freely by the author, so that the frequencies of the co-occurrencies of keywords are relatively lower than that of journals or databases with the lists of available subject headings. This causes some weakness of the system such that it is fatal when a user uses a general term for a specific retrieval, e.g., the keyword "image" is used for many kinds of meanings like "philosophical image" or "visual image". The term "image" corresponds to an vague sets of documents with various meanings. If the indexing system of the database is strict enough, this error will be fixed.

Figure 5.2 shows the distribution of the frequencies of the keywords in the database. The most frequent keyword is "expert system" and it appears 52 times in the database. The number of keywords occurring only once is 1401, twice is 186, three times is 126. It means 90% of keywords occur less than four times in the database and 99% of keywords occur less than 11 times.

Table 5.2 shows the distribution of co-occurrencies of keyword in the database. The second column shows the frequency of co-occurrencies of two different terms, and the third

Document	*List of Keywords*
d_1	object-oriented language, AI programming language, knowledge representation, non-determinism
d_2	knowledge-based system, object model, machine design
d_3	knowledge acquisition, learning, trouble-shooting system, expert system, knowledge extraction rules
d_4	knowledge representation, line drawing interpretation, production system, meta level, certainty factor
$\vdots$	$\vdots$
d_{802}	computer vision, multiagent system, intelligent agent, integration scheme

Table 5.1: Documents from the Journal of Japanese Society for Artificial Intelligence

freq.	$c(x, y)$ $(x \neq y)$	$c(x, y)$ (all)
≥ 6	0	76
5	4	28
4	4	40
3	26	116
2	117	303
1	4 650	6 051
0	1 637 777	1 637 777
total	1 642 578	1 644 391

Table 5.2: Distribution of co-occurrencies

column shows the index of all combinations of the keywords (including the frequency of every terms).

Table 5.3 shows the distribution of size of tolerance classes according to different values of θ from 1 to 4. Each row indicates the numbers of tolerance classes corresponding to their cardinals (number of elements), respectively. For example, when $\theta = 1$, there are totally 96 tolerance classes of terms having only one element, and 85 tolerance classes of terms having two elements, etc. Two last rows indicate the mean (M) and the standard deviation (σ) of sizes of tolerance classes, respectively.

Consider an example for the query $Q = \{t_{19}, t_{234}, t_{235}\}$. With $\theta = 2$, the system produces $\mathcal{U}(\mathcal{R}, Q) = \{t_{11}, t_{19}, t_{160}, t_{203}, t_{234}, t_{235}\}$ and $\mathcal{L}(\mathcal{R}, Q) = \{t_{234}, t_{235}\}$. It gives the answer at three levels $A_{31} = \{d_{81}\}$, $A_{51} = \{d_{363}, d_{798}\}$ and A_{52} with 105 documents. For the same query, if using the Boolean AND operation we obtain only one document $A_{AND} = \{d_{81}\}$, and if using the Boolean OR operation we obtain 12 documents in the same level $A_{OR} = \{d_7, d_{14}, d_{81}, d_{85}, d_{91}, d_{114}, d_{211}, d_{361}, d_{363}, d_{420}, d_{534}, d_{798}\}$.

Other experimental results are shown with more detail in appendix B.

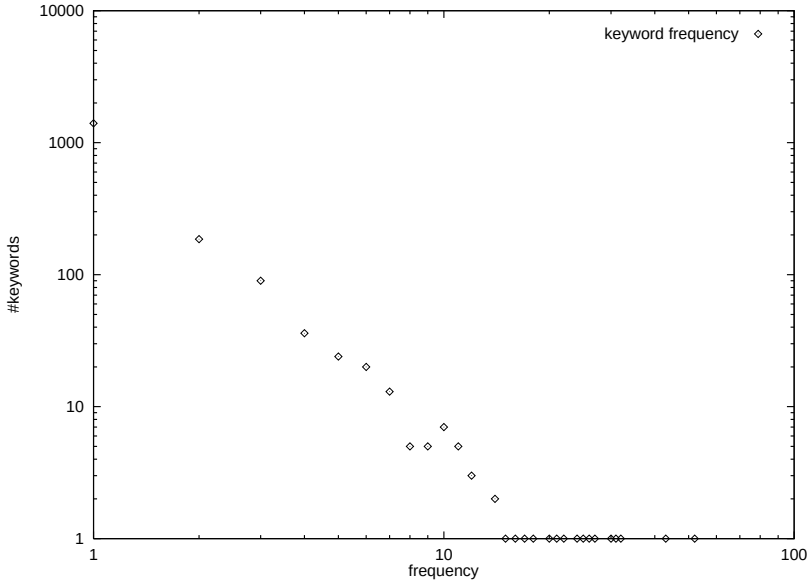

Figure 5.2: Term Frequency in the Database

$\mid I(x) \mid$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$
5	432	3	1	0
4	391	11	3	1
3	176	28	9	1
2	85	114	37	11
1	96	1645	1763	1800
M	6.296	1.667	1.038	1.009
σ	6.569	0.748	0.250	0.115

Table 5.3: Distribution of size of tolerance classes regarding threshold θ

Chapter 6

Evaluation

6.1 Measures of Evaluation

We evaluate the system using the measures of *precision* P and *recall* R which are the best-known measures for evaluating information retrieval systems [16]. For a query, precision is the ratio of relevant documents in the answer set to all documents in the answer set and recall is the ratio of relevant documents in the answer set to the number of all relevant documents in the database. These measures are defined as follows

$$P(Q) = \frac{\mid REL(Q) \cap A \mid}{\mid A \mid} \tag{6.1}$$

$$R(Q) = \frac{\mid REL(Q) \cap A \mid}{\mid REL(Q) \mid} \tag{6.2}$$

where $REL(Q) : \mathcal{Q} \rightarrow \mathcal{P}(\mathcal{D})$ is a set of all relevant documents in the database to the query Q, and $A \in \mathcal{P}(\mathcal{D})$ is union of documents retrieved at the level being considered and documents at all levels ranked before this level regarding the relevance rank. For example, to calculate the precision or recall of the level [2-3] we have $A = A_{11} \cup A_{21} \cup A_{22} \cup A_{23}$. In a large scale database one does not know all elements of $REL(Q)$ but in this testing system we choose manually the relevant documents for a number of queries.

6.2 Discussion

Table 6.1 shows the average of precision and recall for $\theta = 1$ to 4. In this table, the "layer" stands for a group of retrieval levels, for example, the "layer 2" stands for the levels [2-1], [2-2] and [2-3]. And the row "layer 2" shows the precisions and the recalls for the set A regarding the level [2-3] ($A = A_{11} \cup A_{21} \cup A_{22} \cup A_{23}$). The layer 5 is divided into two sublayers called "layer 5 (good)" and "layer 5 (bad)". The "layer 5 (good)" is a sublayer of documents whose values of the secondary ranking function ν is relatively good. The "layer 5 (bad)" is the complement of the "layer 5 (good)" in the layer 5. In most case, the boundary between "good" and "bad" is fixed by this way for all documents. If $\nu > 0$ then the document belongs to "layer 5 (good)", else to "layer 5 (bad)". For example, for

Layer	$\theta = 1$		$\theta = 2$		$\theta = 3$		$\theta = 4$	
	P	R	P	R	P	R	P	R
2	1.00	.031					1.00	.010
3	1.00	.093	.636	.144	.773	.175	1.00	.113
4	.656	.216	.667	.165	.708	.175	.917	.113
5 (good)	.414	.814	.422	.814	.430	.825	.435	.835
5 (bad)	.026	.969	.006	.845	.118	.845	.228	.845

Table 6.1: Retrieval results regarding threshold θ

$\theta = 1$ at level [4-3] or higher levels regarding the relevance rank, the precision average of retrieval is 0.656 and the recall average of retrieval is 0.216. Normally, the higher the precision value the lower the recall value, and vice versa.

The results in Table 6.1 show a number of good points of the method. In layers 2 and 3 the precisions are with high values and the recalls are with low values which show that matching levels before [3-3] (sometimes [4-3]) contain only very relevant documents regarding the relevance rank. In contrast, in layer 5 the recalls are with high values. It means that the system can retrieve almost all the relevant documents. These features allow the user to choose documents with various degrees of satisfaction. If the user wants to find very relevant documents, he/she may take the documents in the layers 2, 3, or 4. If the user want to find all the relevant documents in the database, you can also take more documents from the layer 5.

The value of θ does not influence the precision and the recall in levels of high relevance rank (e.g., 1, 2 and 3), but it influences them at low levels [5-1] and [5-2]. In layer 5, by increasing θ the precision is obviously increased though the recall is not so much decreased. Alternatively, if θ has lower values (1 or 2), almost all relevant documents in the database are retrieved. It means that the user can select the value of θ in order to reduce or enlarge the number of retrieved documents (depending on whether the layer 5 contains too many or too few documents which are somehow less relevant). It is a good strategy for the user to start retrieving with higher threshold θ and after getting the feedback, reduce or raise the value of θ and continue the retrieval process.

The secondary ranking is useful because without it the row "layer 5 (good)" will be deleted and only the "layer 5 (bad)" exists. Rough overlapping strategy may result too many documents and the precisions go down rapidly from the layer 4 to the layer 5 if we do not use the secondary ranking function ν. In our tests, ν divides documents in the layer 5 into several classes ranking from "relatively relevant" to "much less relevant".

There are also several limitations of the method, particularly in the cases when documents or queries contain high frequent terms. High frequent terms have large tolerance classes in comparison with low frequent terms. As the consequence, when the query contains only low frequent terms the system can only retrieve documents with high frequent terms at the level [3-3]. The reason of this defect is in this situation the upper approximations

of such documents are large, and they may also include the small upper approximation of such a query. Alternatively, when the query contains high frequent terms the system retrieves documents only with low frequent terms which often are less relevant. These limitations suggest us some directions for the future work.

6.3 Comparison with Boolean Model

The most main differences between the Boolean retrieval and the rough set retrieval are NOT operation and structured formulations of the query. Though these information retrieval models cannot be compared at this feature, there are some similarities between them. In fact Boolean operations can be applied on the context of set theory on which the rough set model are applied in this work.

We implemented the Boolean AND and OR operations in a system similarly to that of the rough tolerance method. Though we cannot say that this comparison reflects completely the characteristics of these two methods, their application potentials however can be observed.

Method		P	R
Boolean	AND	1.00	.062
	OR	.413	.814
Rough	layer 2	1.00	.010
(average)	layer 3	.797	.131
	layer 4	.707	.168
	layer 5 (good)	.425	.822
	layer 5 (bad)	.057	.876

Table 6.2: Retrieval results by Boolean operations compared with rough sets method

Table 6.2 shows the results of AND and OR retrievals for the same information problem using in the architecture of our method. The precision and recall value of the answer set provided by AND operation is approximately similar to that of rough equalities (layer 2). And the evaluation value of OR operation is approximately similar to that of rough overlaps (layer 5). Rough inclusions (layers 3 and 4) provides the evaluation value between AND and OR operations. It means that the rough inclusions can provide well ordered answers related to the Boolean operations. set of documents larger than by rough inclusions (layers 3 and 4) but the number is smaller than by rough overlaps (layer 5).

The weakness of Boolean model in comparison with the rough set model is that it cannot put all the suitable documents without the keywords in the query. Documents with keywords which are semantically related to the query may fall out (cannot be retrieved). Alternatively, rough overlap strategy can retrieve almost all the relevant documents even if they do not have any keywords used in the query.

6.4 Comparison with Vector Space Model

The vector space model of information retrieval do matches between document and query by the distance between two vectors. Documents in vector space model are represented as vectors of which each element is a term.

The vector space model without term weights can easily be transformed into a context in the information retrieval system defined in chapter 2. Relevance of document i is denoted by $r_i \in \mathbb{R}^+$ and defined as $r_i = \Sigma_j x_{ij} q_j$ where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iM})$ and $\boldsymbol{q} = (q_1, q_2, \ldots, q_M)$ are vectors which represent the documents and the query, respectively [35]. In general, each element in these vectors has real values, i.e., $x_{ij} \in [0, 1]$ and $q_j \in [0, 1]$, and jth element indicates the weight of the term j for the document or query.

For the convenience, in many cases r_i is constrained so that $r_i \in [0, 1]$ by a normalization. The most popular normalization coefficients are $1/(\sqrt{\Sigma_j x_{ij}^2}\sqrt{\Sigma_j q_{ij}^2})$ and $1/(\Sigma_j x_{ij}^2 + \Sigma_j q_{ij}^2 - \Sigma_j x_{ij} q_j)$. The former coefficient is called *cosine measure* defined as follows

$$r_i' \triangleq \frac{\Sigma_j x_{ij} q_j}{\sqrt{\Sigma_j x_{ij}^2}\sqrt{\Sigma_j q_{ij}^2}} = \frac{(x_i, q)}{\mid x_i \mid\mid q \mid} = \cos\theta_{(x_i, q)} \tag{6.3}$$

Consider the case without term weights, i.e., $x_{ij} \in \{0, 1\}$ and $q_j \in \{0, 1\}$. We can re-define $d_i \in \mathcal{T}$ and $Q \in \mathcal{T}$ as follows

$$d_i \triangleq \{t_j \in \mathcal{T} \mid x_{ij} = 1\}, \quad Q \triangleq \{t_j \in \mathcal{T} \mid q_j = 1\} \tag{6.4}$$

According to (6.4), cosine measure is described in the set form [2] as

$$r_i' = \frac{\mid d_i \cap Q \mid}{\mid d_i \mid^{\frac{1}{2}} \times \mid Q \mid^{\frac{1}{2}}} \tag{6.5}$$

The latter coefficient of normalization of the relevant measure is more simple in the style of sets as follows

$$r_i'' \triangleq \frac{\Sigma_j x_{ij} q_j}{\Sigma_j x_{ij}^2 + \Sigma_j q_{ij}^2 - \Sigma_j x_{ij} q_j} = \frac{\mid d_i \cap Q \mid}{\mid d_i \cup Q \mid} \tag{6.6}$$

In general, a rare term characterizes the document more than a frequent term if any term weights are not specified. By the rough sets model in this work, which uses co-occurrencies of keywords, a term with low frequency has a smaller set of approximation class than other terms with high frequency. A query with rare terms may have a non-empty lower approximation more frequently than the query only with frequent terms. The method proposed in this work regards lower approximations as more important than upper approximations and it makes good use of rare terms.

The method of vector space model can be applied to develop the rough set method by integrated in the secondary ranking measure used in section 4.3. These measures are also association measures like (4.12), and used as a secondary ranking function. If the cosine measure is used in the secondary ranking method, it is a combination of the rough set model and the vector space model and it will be an integrated information retrieval model.

Chapter 7

Conclusion

We have presented a rough set based approach to information retrieval in which instead of using an equivalence relation in conventional rough set theory we employ a tolerance relation. We have determined a suitable tolerance space for the information retrieval problem by using co-occurrencies of keywords, and a matching algorithm that is essentially based on rough tolerance inclusions. We believe that the use of tolerance relation in the rough set model is suitable for achieving high retrieval effectiveness. The real-world illustrative example has shown advantages of the method. The preliminary results reported here suggest that some constrains of the method should be more considered with more details, such as the indexing system or system parameters.

When this model is implemented in a system, several points need to be further investigated. In each retrieval, the system starts with a higher value of the threshold θ (by default, $\theta = 4$) then reduce the value of θ according to the user's satisfaction. As the size of the answer set for a high value of threshold is relatively smaller (relatively large with low values), it is easier to deal with high value of θ. After the first trial, if the user does not satisfy with the high value of the threshold and wants to get more documents, he/she may try to retrieve again with a lower value of the threshold θ.

The following directions would be continued to this research.

- Introducing term weights to the model: the term weights are assigned according to the importance of the terms to the documents. The system with weighted terms will be evaluated by comparing to other conceptual models associated with term weights, e.g., fuzzy set model.
- Evaluating the model with large scale databases: in this work a small scale database in a narrow field is used for experiments. The merit of the method will be really proved in a big database. Moreover, if the documents are indexed appropriately (whether it is automatically or manually), probably results of the retrieval will be increased.
- Integrating the model with others: other retrieval models can be mixed with the rough tolerance model. For example, by using the term weights and cosine measure as the secondary ranking, we expect to obtain an integrated retrieval model of rough set model and vector space model.

Acknowledgements

At first, I would like to express my appreciation to Visiting Associate Professor Tu Bao Ho, the advisor who has not only suggested this topic and initial idea of the model, but also taught me the essence of machine learning. I would like to thank Professor Masayuki Kimura, Associate Professor Hiroshi Shimodaira and Professor Milan Vlach of Artificial Intelligence Laboratory who gave me many suggestions and advises, Professor Hiroakira Ono of Computational Logic Laboratory who taught me the essence and the beauty of axiomatic set theory, and Associate Professor Manabu Okumura of Natural Language Processing Laboratory who taught me the basic notions of artificial intelligence.

Members of Artificial Intelligence Laboratory have had many valuable talks with me during one and a half years, particularly Trong Dung Nguyen has contributed many helpful discussions. I also want to thank Shingo Fuyuki, Ritsuko Kobayashi, Shojiro Moribe, Koji Shinoda and Takahiro Uragami for their knowledge of mathematics in the first half year at JAIST. My old colleagues at the University of Library and Information Science, who are the experts of the documents, sometimes gave me many valuable knowledge in the fields of library and information science.

Finally, I express my gratitude to my mother, brothers, sisters, and father who passed away last year, for their supports of my life in universities and Shizue Yamawake who has searched documents from many databases suitably for my research and encouraged me leaving to study at JAIST.

1997年2月 要

Bibliography

[1] N. J. Belkin, W. B. Croft. Retrieval techniques. *Annual Review of Information Science and Technology*, vol.22, 1989, pp. 109–145.

[2] S. K. Bhatia, V. V. Rhagavan. User profiles for information retrieval. Z. W. Ras, M. Zemankova (eds.) *Methodologies for Intelligent Systems: 6th International Symposium (ISMS'91)*. Springer-Verlag, 1991, pp.102–111 (Lecture Notes in Artificial Intelligence 542).

[3] A. Bookstein. Probability and fuzzy-set applications to information retrieval. *Annual Review of Information Science and Technology*, vol.20, 1985, pp.117–151.

[4] H. Chen. Machine learning for information retireval: neural networks, symbolic learning, and genetic algorithms. *Journal of the American Society for Information Science*, vol.46, no.3, 1995, pp.194–216.

[5] T. E. Doszkocs, J. Reggia, X. Lin. Connectionist models and information retrieval. *Annual Review of Information Science and Technology*, vol.25, 1990, pp.209–260.

[6] D. W. Flater, Y. Yesha. Towards flexible distributed information retrieval. N. R. Adam, B. K. Bhargava eds. *Advanced Database Systems*, Springer-Verlag, 1991, pp.259–276 (Lecture Notes in Computer Science 759).

[7] E. Fox, S. Betrabet, M. Koushik, W. Lee. Extended Boolean models. in W. B. Frakes, R. Baeza-Yates. eds. *Information retrieval: data structures & algorithms*. Prentice Hall, New Jersey, 1992, pp.393–418.

[8] W. B. Frakes. Introduction to information storage and retrieval systems. in W. B. Frakes, R. Baeza-Yates eds. *Information retrieval: data structures & algorithms*. Prentice Hall, New Jersey, 1992, pp.1–27.

[9] W. B. Frakes. Stemming algorithms. in: W. B. Frakes., R. Baeza-Yates. eds. *Information retrieval: data structures & algorithms*, Prentice Hall, New Jersey, 1992, pp.131–160.

[10] K. Funakoshi, T. B. Ho. Information retrieval by rough tolerance relation. *The 4th International Workship on Rough Sets, Fuzzy Sets, and Machine Discovery, RSFD'96*, Tokyo, November 1996, pp.31–35.

[11] D. Harman, E. Fox, R. A. Baeza-Yates, W. Lee. Inverted files. W. B. Frakes, R. Baeza-Yates (eds.) *Information retrieval: data structures & Algorithms*. Prentice Hall, New Jersey, 1992, pp.28–43.

[12] T. B. Ho, K. Funakoshi. A tolerance relation based method for information retrieval. *The Joint Pacific Asian Conference on Expert Systems, PACES'97*, Singapore, February 1997 (in press).

[13] P. B. Kantor. Information retrieval techniques. *Annual Review of Information Science and Technology*, vol.29, 1994, pp. 53–90.

[14] M. Koll, P. Srinivasan. Fuzzy versus probabilistic models for user relevance judgments. *Journal of the American Society for Information Science*, vol.41, no.4, 1990, pp.264–271.

[15] E. J. Lemmon. *Introduction to axiomatic set theory*. Routledge & Kegan Paul, 1968. **石本新**, **高橋敬吾訳**. **公理的集合論入門**. **東京図書**, 1972.

[16] S. Miyamoto. *Fuzzy sets in information retrieval and cluster analysis*. Kluwer Academic Publishers, 1990, 259pp.

[17] **中村昭**, **津本周作**, **田中博**, **小林聡**. **ラフ集合理論とその応用**. **人工知能学会誌**, vol,11, no.2, 1996, pp.35–41.

[18] J. Nieminen. Rough tolerance equality and tolerance black boxes. *Fundamenta Informaticae*, vol. 11, 1988, pp.289–296.

[19] C. D. Paice. Evaluation method for stemming algorithms. *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag, 1994, pp.42–50.

[20] Z. Pawlak. *Rough sets: Theoretical aspects of reasoning about data*. Kluwer Academic Publishers, 1991.

[21] Z. Pawlak. Hard and soft sets. W. P. Ziarko (ed). *Proceedings of the International Workshop on Rough Sets and Knowledge Discovery*, Springer-Verlag, 1993, pp.130–135.

[22] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, W. Ziarko. Rough Sets. *Communications of ACM*, vol.38, no.11, 1995, pp.89–95.

[23] L. Polkowski, A. Skowron, J. Zytkow. Rough foundations for rough sets. *The 3rd international workshop on rough sets and soft computing 1994*, pp.142–149.

[24] V. V. Raghavan, R. S. Sharma. A framework and a prototype for intelligent organization of information. *The Canadian Journal of Information Science*, vol.11, 1986, pp. 88–101.

[25] S. R. Ranganathan. *The five lows of library science*. 2nd ed. Asia Publishing House, 1957.

[26] C. J. van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, vol.33, no.2, 1977, pp.106–109.

[27] C. J. van Rijsbergen. *Information retrieval*. 2nd ed. Butterworths, 1979.

[28] E. L. Rissland, J. J. Daniels. Using CBR to drive IR. *Proceedings of International Joint Conference on Artificial Intelligence 1993*, pp.400–407.

[29] *Roget's 21st century thesaurus in dictionary form*. The Philip Lief Group, 1993.

[30] G. Salton, M. J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[31] A. Skowron, J. Stepaniuk. Generalized approximation spaces, *The 3rd International Workshop on Rough Sets and Soft Computing 1994*, pp.156-163.

[32] M. Smail. Case-based information retrieval. S. Wess, K. D. Althoff, M. M. Ritcher (eds). *Topics in case-based reasoning: first European workshop (EWCBR-93): Selected papers*. Springer-Verlag, 1992, pp.404–413 (Lecture Notes in Artificial Intelligence 837).

[33] P. Srinivasan. Intelligent information retrieval using rough set approximations, *Information Processing & Management*, vol.25, no.4, 1989, pp.347–361.

[34] P. Srinivasan. The importance of rough approximations for information retrieval. *International Journal of Man-Machine Studies*, vol.34, no.5, 1991, pp.657–671.

[35] S. Taniguchi. A term dependence model in information retrieval. *Library and Information Science*, no.28, 1990, pp.105–119.

[36] S. Taniguchi. Progress of the research of information retrieval modesl in the 80's : a critical review. *Library and Information Science*, no.30, 1992, pp.59–76.

[37] S. Wartik. Boolean operations. in W. B. Frakes, R. Baeza-Yates eds. *Information retrieval: data structures & algorithms*. Prentice Hall, New Jersey, 1992, pp. 264–292.

[38] S. K. M. Wong, V. V. Raghavan. Vector space model of information retrieval: a reevaluation. C.J.van Rijsbergen ed. *Research and development in information retrieval*, Cambridge University Press, 1984, pp.167–185.

[39] Y. Y. Yao, X. Li, T. T. Lin, Q, Liu. Representation and classification of rough set models, *The 3rd International Workshop on Rough Sets and Soft Computing*, 1994. pp.630-637.

Appendix A

Matching Algorithm in Detail

Matching algorithm described in chapter 4 is re-drawn here. When a set of documents $\mathcal{D}$ and a query Q is inputed, the system calculates the conditions for all $d_j \in \mathcal{D}$ with Q.

Algorithm *Matching*$(Q, \mathcal{D})$
begin
$A_{11} \leftarrow \varnothing, A_{21} \leftarrow \varnothing, \ldots, A_{52} \leftarrow \varnothing$
if $Q \neq \varnothing$ **then**
begin
for $j = 1$ **to** $|\mathcal{D}|$ **do**
begin
if $d_j \neq \varnothing$ **then**
begin
if $Q = d_j$ **then** $A_{11} \leftarrow A_{11} \cup \{d_j\}$ [1-1]Definitably

if $\mathcal{L}(\mathcal{R}, Q) \neq \varnothing$ **then**
if $\mathcal{L}(\mathcal{R}, Q) = \mathcal{L}(\mathcal{R}, d_j)$ **then** $A_{22} \leftarrow A_{22} \cup \{d_j\}$ [2-2]rough bottom equality
if $\mathcal{U}(\mathcal{R}, Q) = \mathcal{U}(\mathcal{R}, d_j)$ **then** $A_{21} \leftarrow A_{21} \cup \{d_j\}$ [2-1]rough equality
if $\mathcal{U}(\mathcal{R}, Q) = \mathcal{U}(\mathcal{R}, d_j)$ **then** $A_{23} \leftarrow A_{23} \cup \{d_j\}$ [2-3]rough top equality

if $\mathcal{L}(\mathcal{R}, Q) \subseteq \mathcal{L}(\mathcal{R}, d_j)$ **then** $A_{32} \leftarrow A_{32} \cup \{d_j\}$ [3-2]rough bottom inclusion
if $\mathcal{U}(\mathcal{R}, Q) \subseteq \mathcal{U}(\mathcal{R}, d_j)$ **then** $A_{31} \leftarrow A_{31} \cup \{d_j\}$ [3-1]rough inclusion
if $\mathcal{U}(\mathcal{R}, Q) \subseteq \mathcal{U}(\mathcal{R}, d_j)$ **then** $A_{33} \leftarrow A_{33} \cup \{d_j\}$ [3-3]rough top inclusion

if $\mathcal{L}(\mathcal{R}, d_j) \neq \varnothing$ **then**
if $\mathcal{L}(\mathcal{R}, d_j) \subseteq \mathcal{L}(\mathcal{R}, Q)$ **then** $A_{42} \leftarrow A_{42} \cup \{d_j\}$ [4-2]rough bottom inclusion
if $\mathcal{U}(\mathcal{R}, d_j) \subseteq \mathcal{U}(\mathcal{R}, Q)$ **then** $A_{41} \leftarrow A_{41} \cup \{d_j\}$ [4-1]rough inclusion
if $\mathcal{U}(\mathcal{R}, d_j) \subseteq \mathcal{U}(\mathcal{R}, Q)$ **then** $A_{43} \leftarrow A_{43} \cup \{d_j\}$ [4-3]rough top inclusion

if $\mathcal{L}(\mathcal{R}, Q) \cap \mathcal{L}(\mathcal{R}, d_j) \neq \varnothing$ **then** $A_{51} \leftarrow A_{51} \cup \{d_j\}$ [5-1]rough bottom overlap
if $\mathcal{U}(\mathcal{R}, Q) \cap \mathcal{U}(\mathcal{R}, d_j) \neq \varnothing$ **then** $A_{52} \leftarrow A_{52} \cup \{d_j\}$ [5-2]rough top overlap
end
end
end
end

Appendix B

An Example of Retrieval in Detail

We show some example retrievals here. Let $Q = \{t_4, t_7\}$, where t_4 is a tag for keyword "learning" and t_7 is a tag for keyword "neural network".

For $\theta = 1$

$$
\begin{aligned}
I_1(t_4) &= \{t_1, t_2, t_3, t_4, t_7, t_8, \ldots, t_{1678}\} \ \text{(69 terms)} \\
I_1(t_7) &= \{t_1, t_2, t_4, t_7, t_{12}, t_{16}, \ldots, t_{1759}\} \ \text{(74 terms)} \\
\mathcal{L}(\mathcal{R}, Q) &= \varnothing \\
\mathcal{U}(\mathcal{R}, Q) &= \{t_1, t_2, t_3, t_4, t_7, t_8, \ldots, t_{1759}\} \ \text{(134 terms)}
\end{aligned}
$$

Following documents are retrieved (document names are ad hoc because the internal names are not necessary).

$$
\begin{aligned}
d_1 &= \{t_4, t_7, t_{541}, t_{542}\} \\
\mathcal{L}(\mathcal{R}, d_1) &= \{t_{541}, t_{542}\} \\
\mathcal{U}(\mathcal{R}, d_1) &= \{t_1, t_2, t_3, t_4, t_7, t_8, \ldots, t_{1759}\} \ \text{(134 terms)}
\end{aligned}
$$

Here so $\mathcal{U}(\mathcal{R}, Q) = \mathcal{U}(\mathcal{R}, d_1)$ that d_1 is retrieved at the level [2-3] and $A_{23} = \{d_1\}$.

$$
\begin{aligned}
d_2 &= \{t_2, t_4, t_7, t_{43}\} \\
\mathcal{L}(\mathcal{R}, d_2) &= \varnothing \\
\mathcal{U}(\mathcal{R}, d_2) &= \{t_1, t_2, t_3, t_4, t_5, t_7, \ldots, t_{1759}\} \ \text{(220 terms)}
\end{aligned}
$$

$$
\begin{aligned}
d_3 &= \{t_4, t_7, t_{52}, t_{1230}, t_{1231}\} \\
\mathcal{L}(\mathcal{R}, d_3) &= \{t_{1230}, t_{1231}\} \\
\mathcal{U}(\mathcal{R}, d_3) &= \{t_1, t_2, t_3, t_4, t_7, t_8, \ldots, t_{1759}\} \ \text{(141 terms)}
\end{aligned}
$$

$$
\begin{aligned}
d_4 &= \{t_4, t_7, t_{133}, t_{1240}, t_{1242}\} \\
\mathcal{L}(\mathcal{R}, d_4) &= \{t_{1240}\} \\
\mathcal{U}(\mathcal{R}, d_4) &= \{t_1, t_2, t_3, t_4, t_7, t_8, \ldots, t_{1759}\} \quad (141 \text{ terms})
\end{aligned}
$$

Here

$$
\mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_2), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_3), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_4)
$$

d_2, d_3, d_4 are retrieved at the level [3-3] and $A_{33} = \{d_2, d_3, d_4\}$. And

$$
\begin{aligned}
d_5 &= \{t_4, t_{480}\} \\
\mathcal{L}(\mathcal{R}, d_5) &= \{t_{480}\} \\
\mathcal{U}(\mathcal{R}, d_5) &= \{t_1, t_2, t_3, t_4, t_7, t_8, \ldots, t_{1678}\} \quad (69 \text{ terms})
\end{aligned}
$$

$$
\begin{aligned}
d_6 &= \{t_7, t_{665}, t_{666}, t_{667}, t_{668}\} \\
\mathcal{L}(\mathcal{R}, d_6) &= \{t_{665}, t_{666}, t_{667}, t_{668}\} \\
\mathcal{U}(\mathcal{R}, d_6) &= \{t_2, t_4, t_7, t_{12}, t_{16}, t_{18}, \ldots, t_{1759}\} \quad (74 \text{ terms})
\end{aligned}
$$

$$
\begin{aligned}
d_7 &= \{t_7, t_{838}, t_{839}, t_{840}, t_{841}, t_{842}\} \\
\mathcal{L}(\mathcal{R}, d_7) &= \{t_{838}, t_{839}, t_{840}, t_{841}, t_{842}\} \\
\mathcal{U}(\mathcal{R}, d_7) &= \{t_2, t_4, t_7, t_{12}, t_{16}, t_{18}, \ldots, t_{1759}\} \quad (74 \text{ terms})
\end{aligned}
$$

$$
\begin{aligned}
d_8 &= \{t_7, t_{1232}\} \\
\mathcal{L}(\mathcal{R}, d_8) &= \{t_{1232}\} \\
\mathcal{U}(\mathcal{R}, d_8) &= \{t_2, t_4, t_7, t_{12}, t_{16}, t_{18}, \ldots, t_{1759}\} \quad (74 \text{ terms})
\end{aligned}
$$

$$
\begin{aligned}
d_9 &= \{t_4, t_{1579}, t_{1580}, t_{1581}\} \\
\mathcal{L}(\mathcal{R}, d_9) &= \{t_{1579}, t_{1580}, t_{1581}\} \\
\mathcal{U}(\mathcal{R}, d_9) &= \{t_1, t_2, t_3, t_4, t_7, t_8, \ldots, t_{1678}\} \quad (69 \text{ terms})
\end{aligned}
$$

Here

$$
\begin{aligned}
\mathcal{U}(\mathcal{R}, d_5) &\subset \mathcal{U}(\mathcal{R}, Q), \quad \mathcal{U}(\mathcal{R}, d_6) \subset \mathcal{U}(\mathcal{R}, Q), \quad \mathcal{U}(\mathcal{R}, d_7) \subset \mathcal{U}(\mathcal{R}, Q), \\
\mathcal{U}(\mathcal{R}, d_8) &\subset \mathcal{U}(\mathcal{R}, Q), \quad \mathcal{U}(\mathcal{R}, d_9) \subset \mathcal{U}(\mathcal{R}, Q)
\end{aligned}
$$

then d_5, d_6, d_7, d_8, d_9 are retrieved at the level [4-3] and $A_{43} = \{d_5, d_6, d_7, d_8, d_9\}$.

At the level [5-2], the number of retrieved documents is too many (601 documents) to describe all of them. So a case is shown.

$$
\begin{aligned}
d_{10} &= \{t_1, t_3, t_4, t_{300}, t_{1079}\} \\
\mathcal{L}(\mathcal{R}, d_{10}) &= \{t_{1079}\} \\
\mathcal{U}(\mathcal{R}, d_{10}) &= \{t_1, t_2, t_3, t_4, t_6, t_7, \ldots, t_{1781}\} \quad (191 \text{ terms})
\end{aligned}
$$

Here so $\mathcal{U}(\mathcal{R}, Q) \cap \mathcal{U}(\mathcal{R}, d_{10}) \neq \varnothing$ that d_{10} is retrieved in level [5-2] and $d_{10} \in A_{52}$.

From all above, the answer set for the query is

$$\begin{aligned} A &= A_{23} \cup A_{33} \cup A_{43} \cup A_{52} \\ &= \{d_1\} \cup \{d_2, d_3, d_4\} \cup \{d_5, d_6, d_7, d_8, d_9\} \cup \{d_{10}, \ldots, d_{610}\} \end{aligned}$$

For $\theta = 2$

$$\begin{aligned} I_2(t_4) &= \{t_2, t_3, t_4, t_7, t_{173}\} \ \text{(5 terms)} \\ I_2(t_7) &= \{t_4, t_7, t_{43}, t_{52}, t_{118}, t_{364}\} \ \text{(6 terms)} \\ \mathcal{L}(\mathcal{R}, Q) &= \varnothing \\ \mathcal{U}(\mathcal{R}, Q) &= \{t_2, t_3, t_4, t_7, t_{43}, t_{52}, t_{118}, t_{173}, t_{364}\} \ \text{(9 terms)} \end{aligned}$$

$$\begin{aligned} d_1 &= \{t_4, t_7, t_{541}, t_{542}\} \\ \mathcal{L}(\mathcal{R}, d_1) &= \{t_{541}, t_{542}\} \\ \mathcal{U}(\mathcal{R}, d_1) &= \{t_2, t_3, t_4, t_7, t_{43}, t_{52}, t_{118}, t_{173}, t_{364}, t_{541}, t_{542}\} \end{aligned}$$

$$\begin{aligned} d_2 &= \{t_2, t_4, t_7, t_{43}\} \\ \mathcal{L}(\mathcal{R}, d_2) &= \varnothing \\ \mathcal{U}(\mathcal{R}, d_2) &= \{t_1, t_2, t_3, t_4, t_5, t_7, \ldots, t_{364}\} \ \text{(20 terms)} \end{aligned}$$

$$\begin{aligned} d_3 &= \{t_4, t_7, t_{52}, t_{1230}, t_{1231}\} \\ \mathcal{L}(\mathcal{R}, d_3) &= \{t_{52}, t_{1230}, t_{1231}\} \\ \mathcal{U}(\mathcal{R}, d_3) &= \{t_2, t_3, t_4, t_7, t_{43}, t_{52}, t_{118}, t_{173}, t_{364}, t_{1230}, t_{1231}\} \end{aligned}$$

$$\begin{aligned} d_4 &= \{t_4, t_7, t_{133}, t_{1240}, t_{1242}\} \\ \mathcal{L}(\mathcal{R}, d_4) &= \{t_{133}, t_{1240}, t_{1242}\} \\ \mathcal{U}(\mathcal{R}, d_4) &= \{t_2, t_3, t_4, t_7, t_{43}, t_{52}, t_{118}, t_{133}, t_{173}, t_{364}, t_{1240}, t_{1242}\} \end{aligned}$$

Here

$$\begin{aligned} \mathcal{U}(\mathcal{R}, Q) &\subset \mathcal{U}(\mathcal{R}, d_1), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_2), \\ \mathcal{U}(\mathcal{R}, Q) &\subset \mathcal{U}(\mathcal{R}, d_3), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_4) \end{aligned}$$

d_1, d_2, d_3, d_4 are retrieved at the level [3-3] and $A_{33} = \{d_1, d_2, d_3, d_4\}$.

$$\begin{aligned} d_{11} &= \{t_7, t_{52}, t_{118}, t_{364}\} \\ \mathcal{L}(\mathcal{R}, d_4) &= \{t_{52}, t_{118}, t_{364}\} \\ \mathcal{U}(\mathcal{R}, d_4) &= \{t_4, t_7, t_{43}, t_{52}, t_{118}, t_{364}\} \end{aligned}$$

Here $\mathcal{U}(\mathcal{R}, d_{11}) \subset \mathcal{U}(\mathcal{R}, Q)$ then d_{11} is retrieved at the level [4-3] and $A_{43} = \{d_{11}\}$.

Similar to the case of $\theta = 1$, the number of retrieved documents at the level [5-2] is too many (260 documents) to describe all of them

$$\begin{aligned} d_{10} &= \{t_1, t_3, t_4, t_{300}, t_{1079}\} \\ \mathcal{L}(\mathcal{R}, d_{10}) &= \{t_{300}, t_{1079}\} \\ \mathcal{U}(\mathcal{R}, d_{10}) &= \{t_1, t_2, t_3, t_4, t_6, t_7, \ldots, t_{1781}\} \quad (211 \text{ terms}) \end{aligned}$$

Here so $\mathcal{U}(\mathcal{R}, Q) \cap \mathcal{U}(\mathcal{R}, d_{10}) \neq \varnothing$ that d_{10} is retrieved in level [5-2] and $d_{10} \in A_{52}$.

From all above, the answer set for the query is

$$\begin{aligned} A &= A_{33} \cup A_{43} \cup A_{52} \\ &= \{d_1, d_2, d_3, d_4\} \cup \{d_5, d_6, d_7, d_8, d_9\} \cup \{d_{10}, \ldots, d_{265}\} \end{aligned}$$

For $\theta = 3$

$$\begin{aligned} I_3(t_4) &= \{t_3, t_4, t_7\} \quad (3 \text{ terms}) \\ I_3(t_7) &= \{t_4, t_7, t_{52}\} \quad (3 \text{ terms}) \\ \mathcal{L}(\mathcal{R}, Q) &= \varnothing \\ \mathcal{U}(\mathcal{R}, Q) &= \{t_3, t_4, t_7, t_{52}\} \quad (4 \text{ terms}) \end{aligned}$$

$$\begin{aligned} d_1 &= \{t_4, t_7, t_{541}, t_{542}\} \\ \mathcal{L}(\mathcal{R}, d_1) &= \{t_{541}, t_{542}\} \\ \mathcal{U}(\mathcal{R}, d_1) &= \{t_3, t_4, t_7, t_{52}, t_{541}, t_{542}\} \end{aligned}$$

$$\begin{aligned} d_2 &= \{t_2, t_4, t_7, t_{43}\} \\ \mathcal{L}(\mathcal{R}, d_2) &= \{t_{43}\} \\ \mathcal{U}(\mathcal{R}, d_2) &= \{t_2, t_3, t_4, t_7, t_{29}, t_{43}, t_{52}, t_{67}, t_{186}\} \end{aligned}$$

$$\begin{aligned} d_3 &= \{t_4, t_7, t_{52}, t_{1230}, t_{1231}\} \\ \mathcal{L}(\mathcal{R}, d_3) &= \{t_7, t_{52}, t_{1230}, t_{1231}\} \\ \mathcal{U}(\mathcal{R}, d_3) &= \{t_3, t_4, t_7, t_{52}, t_{1230}, t_{1231}\} \end{aligned}$$

$$\begin{aligned} d_4 &= \{t_4, t_7, t_{133}, t_{1240}, t_{1242}\} \\ \mathcal{L}(\mathcal{R}, d_4) &= \{t_{133}, t_{1240}, t_{1242}\} \\ \mathcal{U}(\mathcal{R}, d_4) &= \{t_3, t_4, t_7, t_{52}, t_{133}, t_{1240}, t_{1242}\} \end{aligned}$$

$$d_{12} = \{t_4, t_{52}, t_{132}, t_{205}, t_{368}, t_{1138}\}$$

$$
\begin{aligned}
\mathcal{L}(\mathcal{R}, d_{12}) &= \{t_{132}, t_{205}, t_{368}, t_{1138}\} \\
\mathcal{U}(\mathcal{R}, d_{12}) &= \{t_3, t_4, t_7, t_{52}, t_{132}, t_{205}, t_{368}, t_{1138}\}
\end{aligned}
$$

$$
\begin{aligned}
d_{13} &= \{t_1, t_2, t_6, t_7, t_{43}, t_{1233}\} \\
\mathcal{L}(\mathcal{R}, d_{13}) &= \{t_{43}, t_{1233}\} \\
\mathcal{U}(\mathcal{R}, d_{13}) &= \{t_1, t_2, t_3, t_4, t_6, t_7, \ldots, t_{1233}\} \quad (15 \text{ terms})
\end{aligned}
$$

$$
\begin{aligned}
d_{14} &= \{t_1, t_6, t_7, t_{11}, t_{16}, t_{34}, t_{52}\} \\
\mathcal{L}(\mathcal{R}, d_{14}) &= \{t_{34}, t_{52}\} \\
\mathcal{U}(\mathcal{R}, d_{14}) &= \{t_1, t_3, t_4, t_6, t_7, t_9, \ldots, t_{89}\} \quad (15 \text{ terms})
\end{aligned}
$$

$$
\begin{aligned}
d_{15} &= \{t_6, t_7, t_{12}, t_{44}, t_{73}, t_{1457}\} \\
\mathcal{L}(\mathcal{R}, d_{15}) &= \{t_{12}, t_{44}, t_{73}, t_{1457}\} \\
\mathcal{U}(\mathcal{R}, d_{15}) &= \{t_3, t_4, t_6, t_7, t_{12}, t_{44}, t_{52}, t_{73}, t_{88}, t_{1457}\}
\end{aligned}
$$

Here

$$
\begin{aligned}
\mathcal{U}(\mathcal{R}, Q) &\subset \mathcal{U}(\mathcal{R}, d_1), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_2), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_3), \\
\mathcal{U}(\mathcal{R}, Q) &\subset \mathcal{U}(\mathcal{R}, d_4), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_{12}), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_{13}), \\
\mathcal{U}(\mathcal{R}, Q) &\subset \mathcal{U}(\mathcal{R}, d_{14}), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_{15}),
\end{aligned}
$$

$d_1, d_2, d_3, d_4, d_{12}, d_{13}, d_{14}, d_{15}$ are retrieved at the level [3-3] and

$$A_{33} = \{d_1, d_2, d_3, d_4, d_{12}, d_{13}, d_{14}, d_{15}\}$$

Similarly the number of retrieved documents at the level [5-2] is too many (141 documents) to describe all of them

$$
\begin{aligned}
d_{10} &= \{t_1, t_3, t_4, t_{300}, t_{1079}\} \\
\mathcal{L}(\mathcal{R}, d_{10}) &= \{t_{300}, t_{1079}\} \\
\mathcal{U}(\mathcal{R}, d_{10}) &= \{t_1, t_3, t_4, t_6, t_7, t_9, t_{10}, t_{42}, t_{300}, t_{1079}\}
\end{aligned}
$$

Here so $\mathcal{U}(\mathcal{R}, Q) \cap \mathcal{U}(\mathcal{R}, d_{10}) \neq \varnothing$ that d_{10} is retrieved in level [5-2] and $d_{10} \in A_{52}$.

From all above, the answer set for the query is

$$
\begin{aligned}
A &= A_{33} \cup A_{52} \\
&= \{d_1, d_2, d_3, d_4, d_{12}, d13, d14, d15\} \cup \{d_{10}, \ldots, d_{149}\}
\end{aligned}
$$

For $\theta = 4$

$$
\begin{aligned}
I_4(t_4) &= \{t_3, t_4\} \quad (2 \text{ terms}) \\
I_4(t_7) &= \{t_7, t_{52}\} \quad (2 \text{ terms}) \\
\mathcal{L}(\mathcal{R}, Q) &= \varnothing \\
\mathcal{U}(\mathcal{R}, Q) &= \{t_3, t_4, t_7, t_{52}\} \quad (4 \text{ terms})
\end{aligned}
$$

$$
\begin{aligned}
d_1 &= \{t_4, t_7, t_{541}, t_{542}\} \\
\mathcal{L}(\mathcal{R}, d_1) &= \{t_{541}, t_{542}\} \\
\mathcal{U}(\mathcal{R}, d_1) &= \{t_3, t_4, t_7, t_{52}, t_{541}, t_{542}\}
\end{aligned}
$$

$$
\begin{aligned}
d_2 &= \{t_2, t_4, t_7, t_{43}\} \\
\mathcal{L}(\mathcal{R}, d_2) &= \{t_2, t_{43}\} \\
\mathcal{U}(\mathcal{R}, d_2) &= \{t_2, t_3, t_4, t_7, t_{43}, t_{52}\}
\end{aligned}
$$

$$
\begin{aligned}
d_3 &= \{t_4, t_7, t_{52}, t_{1230}, t_{1231}\} \\
\mathcal{L}(\mathcal{R}, d_3) &= \{t_7, t_{52}, t_{1230}, t_{1231}\} \\
\mathcal{U}(\mathcal{R}, d_3) &= \{t_3, t_4, t_7, t_{52}, t_{1230}, t_{1231}\}
\end{aligned}
$$

$$
\begin{aligned}
d_4 &= \{t_4, t_7, t_{133}, t_{1240}, t_{1242}\} \\
\mathcal{L}(\mathcal{R}, d_4) &= \{t_{133}, t_{1240}, t_{1242}\} \\
\mathcal{U}(\mathcal{R}, d_4) &= \{t_3, t_4, t_7, t_{52}, t_{133}, t_{1240}, t_{1242}\}
\end{aligned}
$$

$$
\begin{aligned}
d_{12} &= \{t_4, t_{52}, t_{132}, t_{205}, t_{368}, t_{1138}\} \\
\mathcal{L}(\mathcal{R}, d_{12}) &= \{t_{132}, t_{205}, t_{368}, t_{1138}\} \\
\mathcal{U}(\mathcal{R}, d_{12}) &= \{t_3, t_4, t_7, t_{52}, t_{132}, t_{205}, t_{368}, t_{1138}\}
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{U}(\mathcal{R}, Q) &\subset \mathcal{U}(\mathcal{R}, d_1), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_2), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_3), \\
\mathcal{U}(\mathcal{R}, Q) &\subset \mathcal{U}(\mathcal{R}, d_4), \quad \mathcal{U}(\mathcal{R}, Q) \subset \mathcal{U}(\mathcal{R}, d_{12})
\end{aligned}
$$

$d_1, d_2, d_3, d_4, d_{12}$ are retrieved at the level [3-3] and $A_{33} = \{d_1, d_2, d_3, d_4, d_{12}\}$.

Similarly the number of retrieved documents at the level [5-2] is too many (131 documents) to describe all of them

$$
\begin{aligned}
d_{10} &= \{t_1, t_3, t_4, t_{300}, t_{1079}\} \\
\mathcal{L}(\mathcal{R}, d_{10}) &= \{t_4, t_{300}, t_{1079}\} \\
\mathcal{U}(\mathcal{R}, d_{10}) &= \{t_1, t_3, t_4, t_6, t_{42}, t_{300}, t_{1079}\}
\end{aligned}
$$

Here so $\mathcal{U}(\mathcal{R}, Q) \cap \mathcal{U}(\mathcal{R}, d_{10}) \neq \varnothing$ that d_{10} is retrieved in level [5-2] and $d_{10} \in A_{52}$.

From all above, the answer set for the query is

$$
\begin{aligned}
A &= A_{33} \cup A_{52} \\
&= \{d_1, d_2, d_3, d_4, d_{12}\} \cup \{d_{10}, \ldots, d_{136}\}
\end{aligned}
$$