REDER: REtrieval of Documents based on Evidential Reasoning

Lung Albert Chen

ABSTRACT

Knowledge-based retrieval of information has been proved to enhance the performance of conventional retrieval systems. Systems like RUBRIC (RUle-Based Retrieval of Information by Computer) and KADR (Knowledge-Assisted Document Retrieval) have successfully incorporated the techniques developed in the research of knowledge-based systems to solve the retrieval problem. Current knowledge-based retrieval systems, however, share several common problems. They are designed either for special users or their reasoning mechanism is not general enough. When developed for specific users, they reflect user preference instead of expert knowledge; when the reasoning mechanism is not sufficiently general, an inadequate model could lead to counterexamples. The focus of this research is to design a theoretically sound knowledge-based retrieval system for general users. In this system, expert knowledge serves as a basis for automatic query formulation and query evaluation; furthermore, the entire retrieval of information is considered to be a process of evidential reasoning. The foundation of the reasoning methodology is Dempster-Shafer (D-S) theory and its extension. The new approach subsumes the conventional Boolean system and has many advantages over the other retrieval systems. REDER (REtrieval of Documents based on Evidential Reasoning) is a prototype of such a system and demonstrates a complete process of knowledge-based retrieval of information. The results of a retrieval experiment verifies the performance of this system.

1. Introduction

It is claimed that the current period is known as the information age because more information is generated about more topics than ever before. In this complex world, it is often necessary to collect relevant information in order to carry tasks at hand and to make intelligent decisions. However, large data banks of information collected and stored always make it difficult to find the data needed, and to distinguish relevant from extraneous data. Therefore, computer-aided retrieval of information is widely used in solving this problem [Salton 89].

The retrieval of information is the process of selecting related pieces of information according to the information need specified in a query and with no special format requirement for the information obtained. It differs from the retrieval of data from a database system in that data is retrieved only if it exactly matches the constraints specified in a query. It also differs from the retrieval of knowledge in that knowledge is retrieved through inference and is highly structured [Salton and McGill 83].

The retrieval of related information could play a significant role in bridging the gap between the retrieval of data and the retrieval of knowledge. Such a system locates more related

This work was supported by the NASA Grant NCC-2-275, LLNL Grant 442433-26705 and MICRO Grant 88-094.

data and can be considered as a relaxation of a database management system; on the other hand, a system that obtains more related knowledge may reduce the need for inference and could become the first step in knowledge acquisition [Chen 88a].

1.1. Basic Models for the Retrieval of Information

When the first large-scale system for the retrieval of information was developed in the late 1960s, the Boolean approach was adopted as the basic retrieval strategy. Since then, most of the commercially available retrieval systems are based on the conventional Boolean model; the Dialog system of the Lockheed Information System, Medlars system of the National Library of Medicine, and Stairs system of IBM are examples [Cooper 88b], [Bartschi 85].

The Boolean retrieval model, however, encounters several problems. First, its formulas are unfriendly; a user will often confuse the *AND* with the *OR* in specifying a query. Second, a null output or an overloaded output is usually produced that results in either an empty set or a set of documents too large for the user to finish examining. Third, different emphasis cannot be placed on the descriptors in a query, which, in part, is the reason for the other problems [Cooper 88a].

To resolve these problems, the retrieved documents should be ranked according to their importance to the user and the null output should be eliminated. With the ranked output, the user does not need to study all the documents retrieved but can begin with the highest ranked document and then continue down the list until the information need is satisfied. A threshold on the rank value can be established before retrieval , and the system would then generate a group of documents whose rank values are higher than the threshold. This rank is also called the Retrieval Status Value (RSV).

To obtain the RSV for each retrieved document, several retrieval models have been proposed, among which are the vector space, probabilistic, and fuzzy models described in the following sections.

1.1.1. Vector-Space Model

In the vector-space model, an index record is a list of descriptors each accompanied by a weight, which is a real number assigned by the indexer or is a result of automatic indexing. An input query from the user is also a list of descriptors each associated with a weight.

The interpretation of the weights has an interesting geometric meaning. Each set of descriptors is considered to be a vector in a concept space, and the weights associated with each vector are the coordinates of the vector. The concept space is a multidimensional space spanned by the predefined index terms.

The RSV in the vector-space model reveals the similarity between a document and a query or the closeness between the document and query vectors. One similarity function is the *cosine correlation function* [Salton 71] that measures the cosine of the angle between the query and document vectors. The smaller the angle, the larger the RSV.

The vector-space model has several disadvantages. First, it does not support structured query and, as a result, the descriptors can no longer be combined with the Boolean connectives. Second, because it is impossible for a user to "visualize" the multidimensional concept space, the assignment of weights in the input query is not related directly to the retrieval strategy. Third, all the basic term vectors are assumed to be pairwise orthogonal; otherwise, the mathematical model is meaningless. In a recent work on the generalized vector-space model, this assumption was no longer required [Wong and Yao 87].

1.1.2. Probabilistic Model

In the probabilistic model, the retrieval strategy is based on statistical analysis [Bookstein 85]. Because output is a list of documents ranked in descending order according to the probability of their usefulness to the user, the RSV is the probability that a document is relevant to the input query. This is known as the *probability ranking principle* [Cooper 76], [Robertson 77], [Robertson, Maron and Cooper 82] on which the models of indexing and retrieving and the unified model are based.

The probabilistic model of indexing, also called the "model 1" approach to probabilistic retrieval, applies probability-estimation techniques to document indexing [Maron and Kuhns 60]. This approach focuses on the assignment of the correct weights on the index terms to optimize the probability that this document is retrieved, given a set of sample queries. This analysis relies on the information (or relevance feedback) from other users concerning the individual document.

The "model 2" approach to probabilistic retrieval is also a probabilistic model of retrieving that applies probability-estimation techniques to assign weights placed on descriptors in a query [Robertson and Sparck-Jones 76]. Its goal is to optimize the probability that a group of relevant documents is retrieved, given a query. Similar to model 1, this model also requires relevance feedback but in the form of information from an individual user concerning a set of documents.

The third model, proposed in the early 1980s, unifies the above two models [Robertson, Maron and Cooper 82]. It is also based on the probability ranking principle; however, it requires two sources of relevance-feedback information from an individual user relating to other documents and from other users relating to an individual document.

All of these probabilistic models share common drawbacks. First, it is difficult to assign weights in the input query according to the probabilistic interpretation because the user is isolated from all the statistical information of the system. Second, the relevance-feedback data may not be consistent, especially when the users have varying interests. Third, in order to reduce the computational complexity, assumptions of conditional independence are often applied.

1.1.3. Fuzzy Model

The fuzzy model of retrieval is a quite different approach based on the fuzzy-set theory [Zadeh 65] and focuses on the semantic measurement of the relationship between descriptors and documents. The early work was done by Tahani, Radecki, Buell, Kraft and Bookstein [Tahani 76], [Radecki 76], [Radecki 79], [Buell and Kraft 81b], [Buell and Kraft 81a], [Bookstein 80]. Unlike the retrieval of information in the probabilistic and vector-space models, fuzzy-set theorists acknowledge the appeal of the Boolean approach and attempt to maintain the Boolean framework while increasing its flexibility [Bookstein 85].

In the fuzzy model, each descriptor is associated with a fuzzy set of documents; for a given index term, a membership function expresses the degree/extent to which each document matches to the term. In contrast to conventional Boolean systems, whether a document relates to an index term is a matter of degree. In other words, the interpretation of weights in the fuzzy model is the *degree of membership* which reflects the nature of the semantic relationship of a document to a descriptor. This interpretation also offers the indexer a greater flexibility in the process of indexing.

The Boolean connectives *AND*, *OR*, and *NOT* that combine different descriptors in an information request are preserved, but the basic set operations are governed by fuzzy set operations. Most of the early work adopted the *min/max* model [Zadeh 65] for set operations. Threshold models for any generalized Boolean queries were later introduced [Radecki 79], [Buell and Kraft 81b], [Buell and Kraft 81a]; however, it is difficult to maintain generalized Boolean queries that include relevance weights and thresholds [Buell and Kraft 81b], [Buell and Kraft 81a]. This problem was resolved by making Boolean operators context dependent whereby each operator functions differently on the same sets depending on the context in which the sets are found [Bookstein 80].

By the definitions of a fuzzy set and the membership function, uncertainty is not a factor in fuzzy-set retrieval. This approach, therefore, is not acceptable to those who believe uncertainty is intrinsic to the retrieval process and who adopt the probability-ranking principle.

1.2. Motivation

One common assumption shared by all the retrieval models is that the user is fully responsible for the input query. After the concept of weight is introduced, the user is assumed to understand the interpretation of the weights placed in the query. Because of this limitation, the user must tune the weights several times during the retrieval process before a satisfactory result is achieved. This investigation was motivated by the desire to eliminate the assignment of weights by the user and to obtain them from the knowledge of the experts.

The selection of weights is difficult, as is the choice of correct index terms. The documents are indexed by some relevant terms that may differ, but they could be obtained from the descriptors in a query; however, these documents cannot be retrieved if the query does not extend over all the terms. The knowledge of experts (such as librarians) is required, therefore, to insert the appropriate terms in a query. This knowledge should be incorporated in the retrieval system.

This work also focuses on the use of AI techniques to develop a new retrieval model. This task requires answers to the following questions.

- Considering a query as evidence of the information needs of the user, to what degree of belief should a single document be retrieved?
- Given a concept, what other concepts and subconcepts have been used to describe the concept and to what degree of belief?
- How is the *degree of belief* interpreted; what is the definition of a *weight*?
- How should expert knowledge be represented and how should a conceptual structure that best describes the query be constructed?
- With the conceptual structure and the degree of belief established, what RSV should be assigned to each individual document?

1.3. Advantages of the Knowledge-Based System

The goal of this research was to design a framework for a knowledge-based retrieval system, including automatic query formulation/expansion and an improved method of query evaluation. The new system has the following advantages.

- It is theoretically sound; both the knowledge-based query formulation and evaluation are based on theories of uncertainty reasoning in AI.
- It subsumes the Boolean model and obtains superior results.
- It makes possible the bridging of the probabilistic and fuzzy retrieval models and demonstrates that these models are compatible.

1.4. Outline

Section 2 examines several knowledge-based retrieval systems. It also analyzes the basic elements of such systems and those of automatic query formulation and evaluation.

Section 3 describes the general framework of the new knowledge-based retrieval system. The interpretation of *degree of belief*, representation of expert knowledge, format of the general rules, basic operations, and the entire process of retrieval of information are investigated.

A prototype of the new retrieval system is presented in Section 4, including a friendly user interface for the query input and graphic representations of the process of query formulation and evaluation. The issue of performance analysis is also discussed and an experiment result is presented.

Section 5 summarizes the results of this investigation and recommends additional areas of research.

2. Knowledge-Based Retrieval of Information

An example of knowledge-based retrieval system is RUBRIC [Tong et al. 83], [Tong et al. 85] (RUle-Based Retrieval of Information by Computer), which explored the possibility of using AI techniques, especially those for knowledge-based systems, in building a retrieval system. It also generates a query structure according to the knowledge of the user. *Evidential reasoning* in the retrieval of information was first introduced in RUBRIC; here, documents are considered as sources providing evidence that they are related to the input query.

Another example is KADR [Biswas et al. 87] (Knowledge-Assisted Document Retrieval) that extends the Boolean query of the user according to expert knowledge. It not only focused on expanding the query of the user according to expert knowledge but also applied a formal theory for the process of evidential reasoning to retrieve information.

Following are the common features of these two systems

- expert knowledge to formulate queries
- fuzzy-set concept to represent the documents
- inference to combine all the pieces of expert knowledge
- evidential reasoning to cope with imprecise/uncertain knowledge
- retrieval result ranked according to degree of relevance

These two systems are related to this work and are described in more detail in the following sections.

2.1. RUBRIC

RUBRIC is a full text retrieval system developed by Tong [Tong et al. 83], [Tong et al. 85]. It is a rule-based system wherein rules are used to represent the knowledge (preference) of the user in retrieving information. A conceptual hierarchy is the result of the chaining of rules, and this conceptual structure matches each document to determine the RSV of each.

Given a single concept that describes the information need of a user, the manner in which RUBRIC generates a conceptual structure in response to a given concept is based on the pieces of knowledge provided by the user. As in general knowledge-based systems, such information is represented in the form of a rule, and these rules combined are interpreted as a hierarchy of retrieval concepts and subconcepts. By naming a single concept, therefore, the user automatically initiates a goal-oriented search of the tree created by all of the subconcepts used to explain that concept. The subconcepts at the lowest level are further defined in terms of pattern expressions in a text-reference language (contextual terms). The entire structure is then used to assign a RSV to each document.

Assuming, for example, that the user is interested in retrieving all the information relating to the 1989 Baseball World Series , the conceptual hierarchy for this query would then appear in the form of Figure 2.1. Each arc in this structure represents a rule, and has an attached relevance value such that, according to their relevance values, the intermediate topics and keyword expressions contribute to the overall relevance of the document to the root topic. Unlabeled arcs have an relevance value of 1.0 and those representing the conjunctions of an *AND* expression are linked together near their common base. Here, the relevance value and the weights in fuzzy-set models are the same, which is a subjective semantic-based similarity measure. Two rules applied to generating this structure are

$$\textit{Baseball Championship} \rightarrow \textit{Event} \ (0.9)$$

$$\textit{Ball} \rightarrow \textit{Baseball} \ (0.5)$$

where the first rule interpreted as "Baseball Championship" *implies* "Event" to the degree of 0.9, and the second as "Ball" is *evidence* of "Baseball" to the degree of 0.5.

Evidential reasoning in the retrieval of information was introduced in RUBRIC. It begins by recognizing a document as the source of "evidence" on which the system determines the relevance value of that document to the retrieval request. For example, regarding the query structure in Figure 2.1, if a document contains the words "ball," "baseball," and "championship," but no other words referred to the example knowledge base, the leaf nodes of "ball," "baseball," and "championship" all will have a value of 1.0, which is strong evidence that this document is relevant to these concepts. All other nodes have a value of 0.0 which indicates that there is no evidence of the relevance of this document to these concepts.

The relevance values at the leaf nodes are then propagated across the rules toward the root concept and, as a result, the nodes of the tree would be assigned the values shown in the parentheses. In this example, the overall relevance measure of the document to the query is 0.63. The propagation of these values is governed by the similarity measure on the ark; here, in this example, a calculus that models conjunction with the minimum operation, disjunction with the maximum operation, and the arithmetic product as the detachment operator is chosen to propagate the relevance values. The details of these operations lead toward the selection of uncertainty calculi [Tong and Shapiro 85], [Bonissone and Decker 85].

The RUBRIC system is a successful example of an application of ideas from AI in the development of a computer-based system for the retrieval of information; the system is fully implemented and commercially available. It has the following advantages.

- Matching is performed over the whole document.
- The RSV is a relevance value in the [0, 1] range.
- Queries are expressed in a language of rules that enables the user to develop hierarchical knowledge structures of retrieval concepts.
- The users are provided with a collection of graphic tools.

One problem is that RUBRIC assumes that a user will conceptualize a query consistently over time, which is not always a valid assumption. Another problem is that RUBRIC assumes

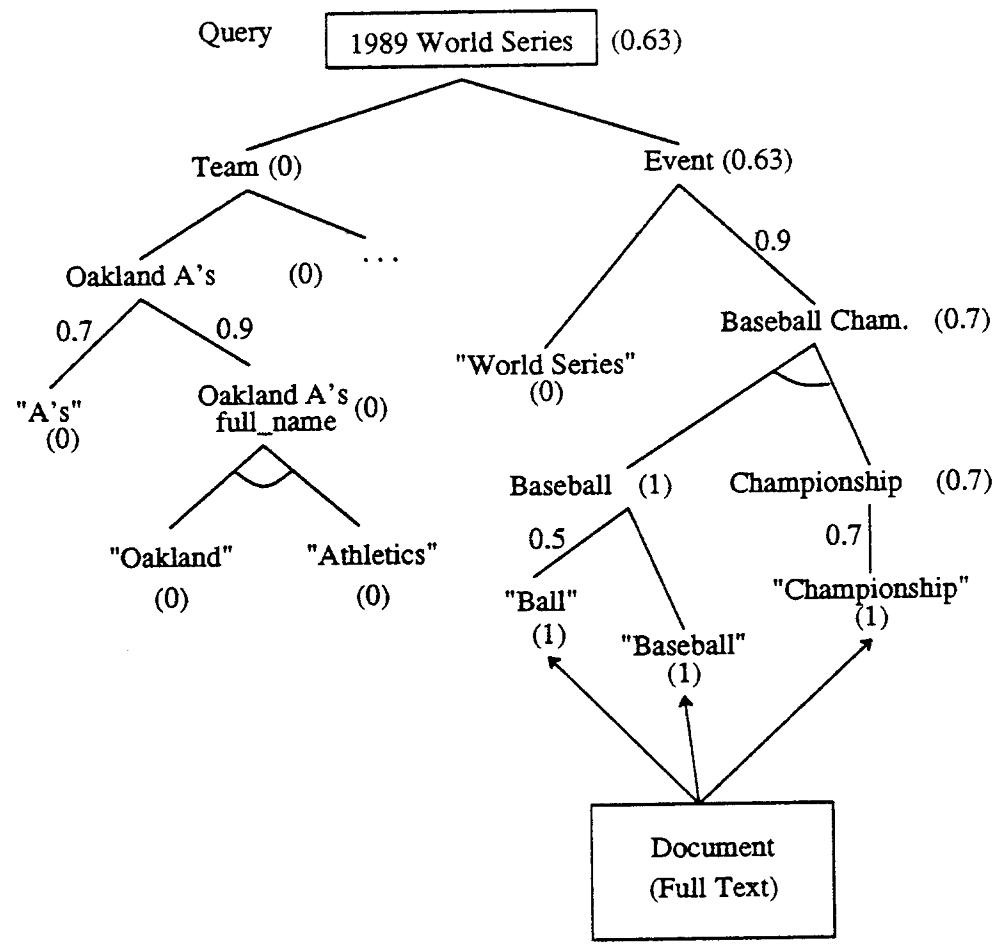


Figure 2.1. Example query in RUBRIC.

that all users are knowledgeable about the domain in which they are interested and about the type of documents they want to retrieve; many fail to reach this level of sophistication.

2.2. Evidential Reasoning and D-S Theory

The D-S theory is the theoretical basis for KADR. The theory was first introduced by Dempster [Dempster 67] and later extended by Shafer [Shafer 76], [Shafer 81], [Shafer 82], [Shafer 87]. A coherent D-S approach to acquiring pieces of evidence relating to hypotheses groups is called "evidential reasoning" [Lowrance and Garvey 82], [Lowrance 82].

The D-S theory is based on the concept of lower and upper probabilities induced by multivalued mapping [Dempster 67], [Yen 86]. Multivalued mapping Γ from an evidence space E to a hypothesis space Θ associates each element in E with a set of elements in Θ ($\Gamma : E \rightarrow 2^{\Theta}$). The element/subset compatibility relationship is denoted by $:\rightarrow$. Given a multivalued mapping and a probability distribution of space E, a basic probability assignment (bpa) of Θ, denoted by $m: 2^{\Theta} \rightarrow [0,1]$, is induced. The bpa is also called mass assignment, mass distribution, or granular distribution. The basic probability value of a subset F of space Θ is

$$m(F) = \sum_{t_i :\rightarrow F} P(t_i) \tag{2.1}$$

where $P(t_i)$ is the probability judgement over $t_i \in E$. The subset F is also called the *focal element*, and Θ is the *frame of discernment*. A legal bpa must have the following properties:

$$\sum_{F \subseteq \Theta} m(F) = 1, \quad m(\varnothing) = 0 \tag{2.2}$$

The probability distribution of Θ is constrained by the bpa. An interval $[Bel(F), Pls(F)]$ is used to measure the degree to which F is believed, which is an arbitrary set in Θ. Here, the lower bound $Bel(F)$ denotes the *belief* of F that counts the degree of belief *necessarily* committed to F, whereas the upper bound $Pls(F)$ is the *plausibility* of F that expresses the degree of belief *possibly* committed to F. Mathematically, these belief and plausibility functions are defined as

$$Bel(F) = \sum_{X \subseteq F} m(X) \tag{2.3}$$

$$Pls(F) = \sum_{X \cap F \neq \varnothing} m(X) \tag{2.4}$$

It should be noted that $Bel(F) = 1 - Pls(\bar{F}) \leq Pls(F)$, $Bel(F) + Bel(\bar{F}) \leq 1$, and the interval $[Bel(F), Pls(F)]$ will reduce to a point-wise Bayesian probability $P(F)$ if all the focal elements are singletons (no composite set has a probability > 0). In this context, $Bel(F) + Bel(\bar{F}) = 1$ [Shafer 76].

If m_1 and m_2 are two basic probability assignments induced from two independent evidential sources, a third bpa $m(C)$ expresses the pooling of evidence from the two sources, and can be computed by Dempster's rule of combination,

$$m(C) = (m_1 \oplus m_2)(C) = \frac{\sum_{A_i \cap B_j = C} m_1(A_i)\, m_2(B_j)}{1 - \sum_{A_i \cap B_j = \varnothing} m_1(A_i)\, m_2(B_j)} \tag{2.5}$$

where A_i, B_j, and C are focal elements in Θ.

Dempster's rule of combination normalizes the intersection of the bodies of evidence from the two sources by the amount of *nonconflictive evidence* between the sources. This amount is represented in the denominator. Eq. (2.5) is also expressed as

$$m(C) = (m_1 \oplus m_2)(C) = K \cdot \sum_{A_i \cap B_j = C} m_1(A_i)\, m_2(B_j) \tag{2.6}$$

where K is the normalization factor with a value of

$$K = \frac{1}{1 - \sum_{A_i \cap B_j = \varnothing} m_1(A_i)\, m_2(B_j)}$$

The D-S theory has the following advantages over other approaches [Yen 86], [Liu 87], [Gordon and Shortliffe 84].

- It facilitates a coherent expression of *information ignorance*. The commitment of belief in a focal element does not imply commitment of the remaining belief to its negation; part of the belief can be reserved to the "don't-know" choice.
- Evidence may relate to groups of hypotheses and be combined coherently. It is able to model the narrowing of the hypothesis set with the accumulation of evidence.
- It subsumes the Bayesian theory in that it can be reduced to the Bayesian. The D-S theory is "probability-related," whereas the Bayesian is "probability-based" and, as a result the D-S does not require full information concerning probabilities.

2.3. KADR

Another knowledge-based retrieval system is the KADR. Unlike RUBRIC, KADR is based on a natural language-processing technique in query formulation; however, the internal representation of the query remains simple Boolean. Because KADR proposes a general evidential-reasoning model for determining the RSV of a document, this section focuses on the evidential reasoning in KADR.

Evidential reasoning in KADR determines the degree of relevance of a given document to a specified concept. Here, concepts are considered as index terms. A document is represented as a fuzzy set of concepts wherein each concept is attached with a degree of membership,

$$d = \{(c_{di}, \mu_d(c_{di}))\} \quad i = 1, \ldots, n$$

where d is a document, c_{di} is a concept contained in this document, and μ_d is the characteristic function of d that maps from the concept space to the range [0,1]. In KADR, the relationship between concepts is represented as a fuzzy measure, called *implication relation*, and is assigned by an expert. An implication from c_i to c_j can take the form of

$$I(c_i, c_j) = \mu_I(c_i, c_j)$$

where μ_I measures the degree of relevance for a given pair of concepts. A concept is relevant to itself to the degree of 1.0.

These two measures determine the mass assignment in KADR. For each concept in a document or each member in the fuzzy set associated with a given document, an evidence space E_i is constructed for that concept. For a given E_i, the mapping between it and the concept space C is illustrated in Figure 2.2 where E_i on the left contains concept c_{di}. A mapping is constructed between c_{di} and each concept c_k implied by it. If there is no such c_k, a mapping is constructed from c_{di} to each c_k that implies c_{di}. The mass assignment associated with the mapping between c_{di} and c_k is defined as

$$m_i(\{c_k\}) = \mu_d(c_{di})\, I(c_{di}, c_k) \tag{2.7}$$

in which m_i is the mass assignment of the evidence space associated with the c_{di}. Knowing all the mappings from c_{di} to C, the ignorance becomes

$$m_i(\Theta) = 1 - \sum m_i(\{c_k\}) \quad c_{di} \rightarrow c_k \tag{2.8}$$

where Θ (here, C) is the hypothesis space and $\rightarrow$ denotes *implies* or *implied by*. This measure is represented by a mapping from a pseudo node p_i in E_i to the entire concept space C.

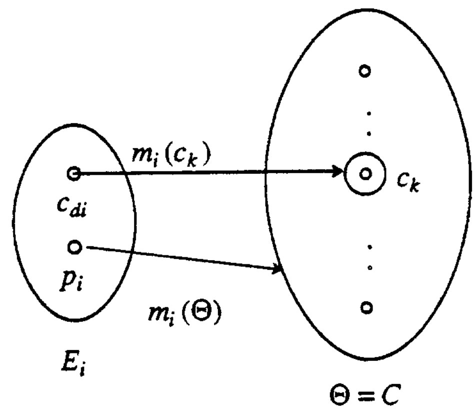


Figure 2.2. Mass assignment in KADR.

Given all the evidence spaces and the mappings between them and C, the measure of all the E_i are combined according to Dempster's rule. Each concept in C has a derived mass assignment, which is the measure specifying to what degree the document is relevant to the concept,

$$r(d, c_k) = m(c_k) = (m_1 \oplus m_2 \oplus \ldots \oplus m_n)(c_k) \tag{2.9}$$

where r is the relevance function of a document to a concept, n is the number of concepts in d, and $\oplus$ is the combining operator in D-S theory. When a query is expressed as a Boolean structure of the concepts in C, the relevance values of the concepts are combined by the geometric-mean and linear interpolation functions corresponding to the *and* and *or* operators as

$$v(c_i \; and \; c_j) = [v(c_i)\,(c_j)]^{1/2} \tag{2.10}$$

$$v(c_i \; or \; c_j) = v(c_i) + [1 - v(c_i)]\, v(c_j) \tag{2.11}$$

where c_i and c_j are concepts in the query. With these functions and given a document and a Boolean query (the internal representation of a query in KADR), the overall relevance value can be calculated.

Unfortunately, the evidential-reasoning model in KADR is not always adequate. First, the ignorance measure in Eq. (2.7) is not always a positive number, and a negative measure conflicts with the original D-S theory; second, KADR does not subsume the Boolean system [Chen 88b].

3. Evidential Reasoning System for the Retrieval of Information

The general framework of evidential reasoning for the knowledge-based retrieval of information proposed in this research is introduced here. In this system, the automatic query formulation is based on the expert knowledge, instead of user knowledge (preference), as in RUBRIC. For query evaluation, all the pieces of partial information are combined in a consistent manner. In Shafer's term, the overall measure of the degree of belief between the query structure and a single document is assigned to that document as its RSV.

As illustrated in Figure 3.1, this new system begins with a query Q submitted by the user. The query is then expanded automatically by the system, based on expert knowledge, to form a conceptual hierarchy (tree). To the right of the hierarchy are the index and document spaces containing all the index terms and the documents, respectively. The terms c_k at the bottom of the hierarchy are used as index terms I_k to obtain the associated set of documents D_k.

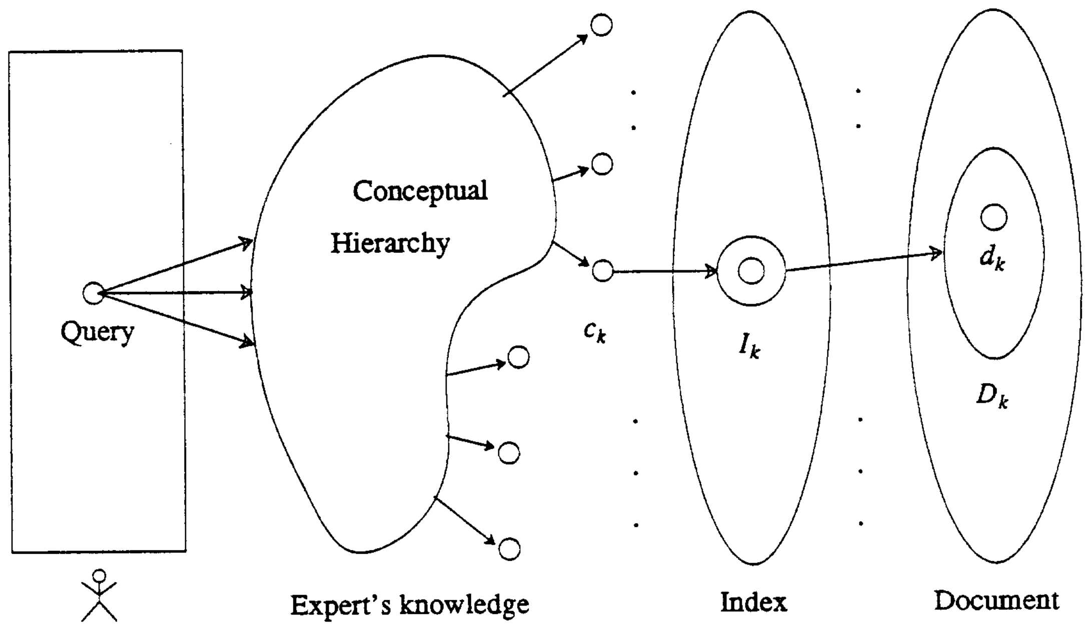


Figure 3.1. Overview of the new retrieval system.

The entire retrieval procedure is a process of evidential reasoning. On each link in Figure 3.1 is attached a belief measure in the range [0,1] whose value is determined or derived by expert knowledge; if not given, the default value is 1. At the final stage, the belief measures induced on the subset of documents are combined, according to the evidence aggregation mechanism introduced later, and the RSV for each document is assigned accordingly.

Because the entire retrieval procedure is a process of evidential reasoning, the RSV is the *belief* according to expert knowledge, that a document is relevant to the input query. This is the *belief ranking principle* on which this investigation is based.

Following is a summary of the different types of rules in the model and the property of the attached belief measure whose source is indicated in the parentheses:

$$Q \quad (user)$$
$$c_k \rightarrow I_{k,}\ (1.0) \quad (default)$$
$$I_k \rightarrow D_{k,}\ (1.0) \quad (indexer)$$
$$c_i \rightarrow c_{j,}\ (v),\ v:[0,1] \quad (expert)$$

As a result,

$$RSV(d_k) = Pls(d_k) \tag{3.1}$$

where c_i and c_j are subconcepts in the conceptual hierarchy, and the attached value v between them is obtained by the expert.

3.1. Automatic Query Formulation

The conceptual hierarchy is constructed between the query of the user and the set of index terms generated by the system. It is the result of automatic query formulation which is a process of chaining together all the rules given by the expert according to the interpretation described in Figure 3.1.

The construction of the conceptual hierarchy is very similar to the one for RUBRIC; however, major differences exist between them. First, in the new system, the hierarchy is based on expert knowledge instead of user preference. Although preference may better reflect the required information, it prevents the general users from accessing the system and, therefore, it is necessary to generate a conceptual structure based on expert knowledge. As a result, a user needs only to choose the concept that best describes the information required through a friendly user interface, and all the work is then left to the system. It is believed that expert knowledge should produce a more consistent and better retrieval result.

Second, the conceptual hierarchy is a result of the *forward chaining* instead of *backward chaining* of the rules established by the expert. In RUBRIC, a concept is placed on the left-hand side of a rule, and a concept implied by a subconcept is placed on the right. Given a concept, all the subconcepts that imply this concept are fetched by matching them with the right-hand side of the rule. The interpretation of these rules is that this concept is defined or described by the subconcepts. The approach followed is to place the concept on the left-hand side and the subconcept on the right. Given a query concept, for example, all the subconcepts used to describe this concept can be obtained by matching the concept with the left-hand side of each rule. A major advantage of forward chaining is that it not only expands a query concept into a conceptual hierarchy but also expands each subconcept on the leaf nodes into a set of documents.

Example 3.1 (Forward Chaining).

Suppose a user is interested in *AI* $(Q = AI)$, and is directed to the subconcepts *Natural Language (NL)* and *Expert System (ES)* by the rules

$$AI \rightarrow NL\ (0.3)$$
$$AI \rightarrow ES\ (0.5)$$

and the subconcept *Expert System (ES)* is further explored by the subconcepts *Knowledge Representation (KR)* and *Reasoning (RE)* as encoded in the following rules

$$ES \rightarrow KR \ (0.5)$$

$$ES \rightarrow RE \ (0.5)$$

Then the forward chaining described above will produce a conceptual hierarchy mapping from general concepts *(AI)* to specific subconcepts *(KR, RE, NL)*, and from subconcepts to sets of documents. This is displayed in Figure 3.2, where d_i stands for a document, and a dashed link represents an index rule. Some intermediate chaining steps following the subconcepts are skipped.

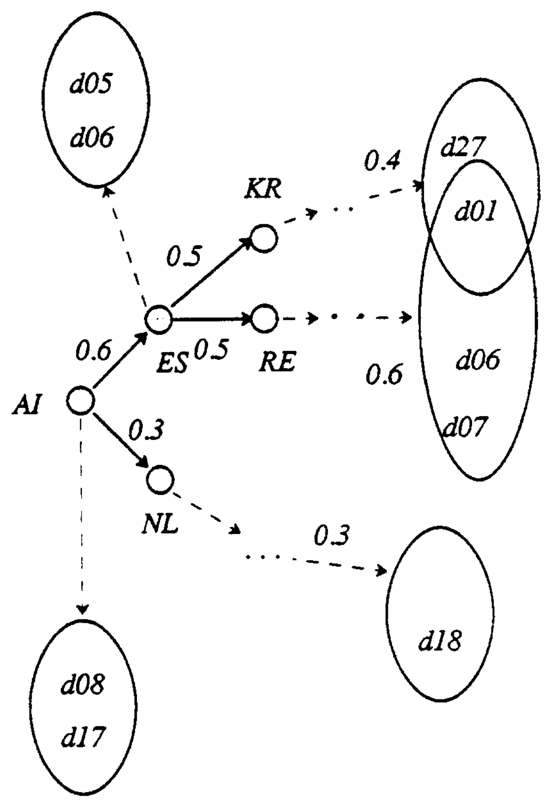


Figure 3.2. Example of forward chaining.

Third, *query* instead of *document* is treated as evidence which indicates that query is evidence of a user's information need and that the conceptual structure is the evidence in determining which document should be selected. This approach differs from RUBRIC and is possible only when the rules and index mapping have the same direction. The advantage is that a single user concept triggers the entire process of forward chaining or evidential reasoning in the sequel; no other control mechanism is necessary. This difference is illustrated in Figure 3.3.

3.2. Interpretation of Rule Certainty

In this investigation, rule certainty in this research is an *experience-based belief measure* based on the extended D-S theory [Liu 87], [Chen 88b]. Given a rule between a concept c_i and a sub-concept c_j, the expert is asked to what degree of belief will c_j be selected to define c_i or to

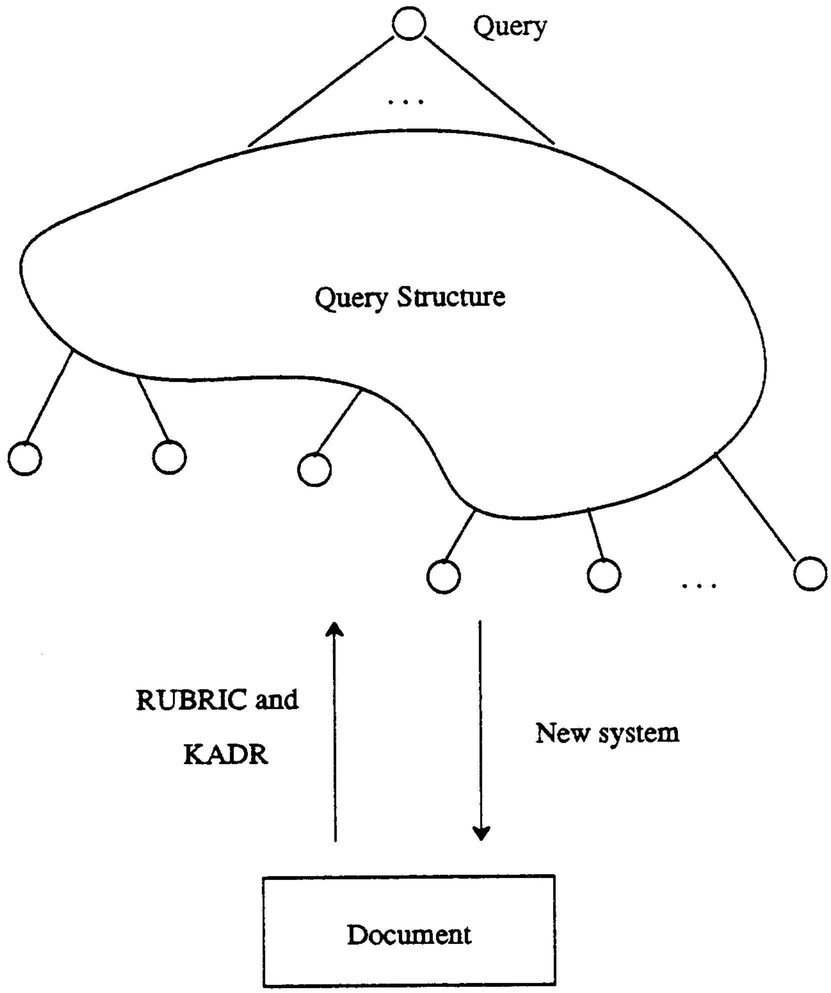


Figure 3.3. Reasoning directions in RUBRIC and in the new retrieval system.

what degree of belief has c_j been used by others to describe c_i. Here, the terms *degree of belief* and *conditional mass assignment* are interchangeable, and the *degree of belief* differs from the *belief measure* of a focal element which is equal to the degree of belief *necessarily* committed to that element.

Because a given rule summarizes an expert's knowledge and experience, rule certainty is interpreted accordingly. Given a concept c_i and a subconcept c_j as illustrated in Figure 3.4, the first task of the expert is to obtain a list of pieces of knowledge $(c_{j1}, c_{j2}, \ldots, c_{jn})$ used to describe c_i, and are semantically compatible with c_j. This knowledge can be classified into many categories, such as definition of c_i, information in the text books, conversation with other experts, graphic analogy, and the expert's subjective opinion, and so on.

Example 3.2 (Knowledge Acquisition).

Suppose a new rule is added to an existing knowledge-based retrieval system. The purpose of this rule is to direct the user who is interested in the concept *Knowledge-Based Retrieval of Information (KBRI)* to the subconcept *Evidential Reasoning in Retrieval of Information (ERRI)*. As displayed in Figure 3.5. the subconcept *ERRI* is supported by a background frame containing

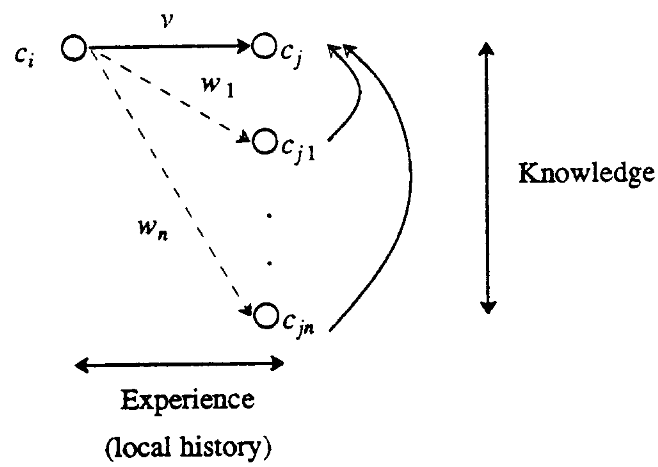


Figure 3.4. Rule certainty.

related research (RUBRIC, KADR and this work) and the strength of this rule depends on how well the background frame interprets the concept *KBRI*, according to expert knowledge.

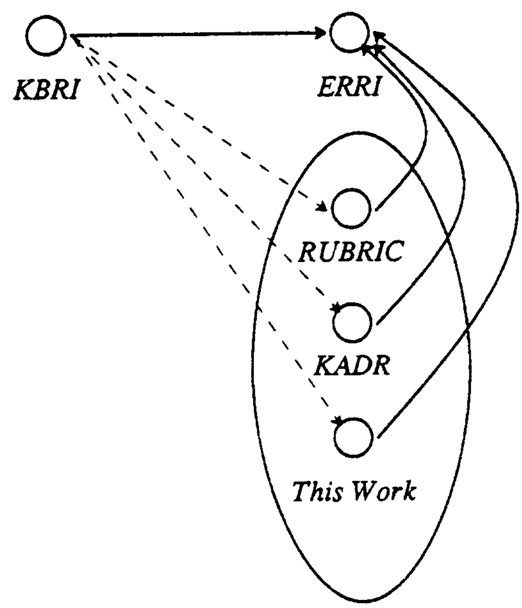


Figure 3.5. Example of knowledge acquisition.

There are very few limitations on the manner in which an expert describes the concept and assigns weights; however the process of obtaining this knowledge is a problem of knowledge acquisition [Feigenbaum 77], [Waterman and Hayes-Roth 82], [Boose 85] and is not directly related to this investigation.

The second task is to assign a weight to each piece of knowledge based on the experience of the expert. Weight w_k is assigned to knowledge c_{jk} to reflect its relevant importance. Weight w_j is a number in the range [0, 1] and $w_1 + w_2 \ldots + w_n = 1$. In addition, w_j approximates the conditional probability $P(c_{jk}/c_i)$ which is considered to be local history independent of the other pieces of knowledge. Given the weights, the overall degree of belief v of this rule takes the form of

$$v = m(c_j/c_i) = \sum_k w_k \tag{3.2}$$

By defining rule certainty as a conditional mass assignment, the equation for the chaining of rules followed; to determine the degree of belief by which c_i is described by subconcept c_j and with some intermediate subconcepts c_k between c_i and c_j, the measure between c_i and c_j is equal to the summation of the pairwise multiplication of the conditional mass $m(c_j/c_k)$ and $m(c_k/c_i)$ as

$$m(c_j/c_i) = \sum_k m(c_j/c_k)\, m(c_k/c_i) \tag{3.3}$$

Example 3.3 (Chaining of Rules).

Suppose the concepts *AI, Natural Language (NL), Expert System (ES), and Knowledge Representation (KR)* are linked as in Figure 3.6. Each link stands for a rule and the number attached to a link represents the strength (conditional mass) of the rule.

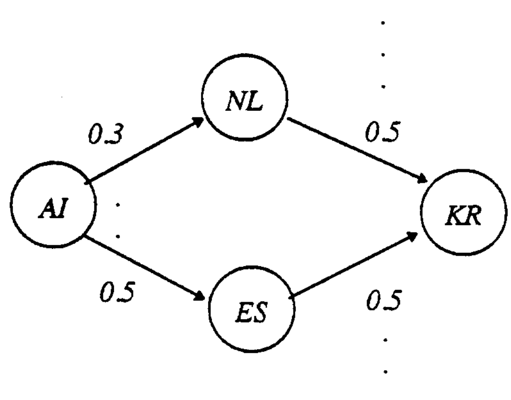


Figure 3.6. Example of chaining of rules.

According to Eq. (3.3), then, the conditional mass assignment of *KR* given *AI* is

$$\begin{aligned} m(KR/AI) &= m(KR/NL)\, m(NL/AI) + m(KR/ES)\, m(ES/AI) \\ &= 0.5 \times 0.3 + 0.5 \times 0.5 \\ &= 0.4 \end{aligned}$$

3.3. Evidence Aggregation

All the rules in the system cover the process of retrieval of information. Beginning with a query (which is a concept chosen by a user) and on through the conceptual hierarchy and then from the leaf nodes to the textual terms and from the index terms to the sets of documents, the entire path is linked by rules. Rule uncertainty is then defined as conditional mass assignment. This process, therefore, can be considered as a process of evidential reasoning.

The *indexing rules* that map an index term into a set of documents and the *default rules* that map a subconcept at the leaf node into an index term are special examples of rule-certainty measurement. These rules are either unconditional or conditional having the conditional mass assignment equal to 1. The default rule is equivalent to strong evidential reasoning in RUBRIC.

3.3.1. Overview of Evidence Aggregation

The RSV is calculated in the document space based on the information in the conceptual hierarchy. In Figure 3.7, to the left of the dashed line is the expanded query showing the conceptual hierarchy and to the right is the image of this hierarchy defined in terms of a set of documents (the node of a subconcept is replaced by a set of documents). Here, D_i denotes the document subspace corresponding to the subconcept c_i. All the uncertainty measures on the links of the conceptual hierarchy are transfered to the document-space hierarchy.

The construction of the document-space hierarchy begins with the leaf nodes of the conceptual hierarchy. Each node is associated with a set of documents indexed by the index term (or subconcept) at that leaf node. The initial value of the mass assignment for the set of documents at the nodes is 1.0, and no discounting is necessary because these nodes are governed by indexing rules. In this example, D_2, D_4, D_5, and D_6 all have an initial value of 1.0.

After setting the initial value, the mass assignment is propagated to the root of the hierarchy (a set of all the documents to be retrieved). The RSV of each document is determined by the *plausibility* measure of each document, which is result of applying D-S theory over the document space at the root. Mass-distribution propagation is described in the following section.

3.3.2. Basic Operations

The basic operations of mass-distribution propagation combine the mass value of the sets at the left (lower level of the hierarchy) into the mass value at the right (higher level). This combination is controlled by the corresponding node in the query structure -- the operation between the nodes corresponding to the sets at the left side. Normally, more than two sets are to be combined. Because the overall value is not affected by the order in which they are combined, the operation (OP) is considered as a binary operation.

In Figure 3.8, the section of the document hierarchy on the right corresponds to the conceptual hierarchy on the left; the dashed links between the leaf nodes represent this relationship. Here, concept c_k is described by two subconcepts c_i and c_j, and the rule uncertainty is measured by v_i and v_j, respectively. The set of documents associated with the subconcept c_i is denoted by D_i and also with D_j and D_k. The uncertainty measure between these sets is transfered from the conceptual structure on the left. It should be noted that D_i and D_j may have some common elements; they appear in individual ovals for clarity. Combining D_i, D_j, and their uncertainty measure, into D_k and its uncertainty measure is discussed below.

If OP is an OR operation, then D_k is equal to the union of D_i and D_j. For any focal element X that has a nonzero mass measure in either D_i or D_j, the derived mass measure of X over space D_k is equal to the summation of the mass measure of X over D_i, $m_i(X)$ and that of X over D_j, $m_j(X)$ discounted by the uncertainty on the links v_i and v_j, respectively. These operations are described by

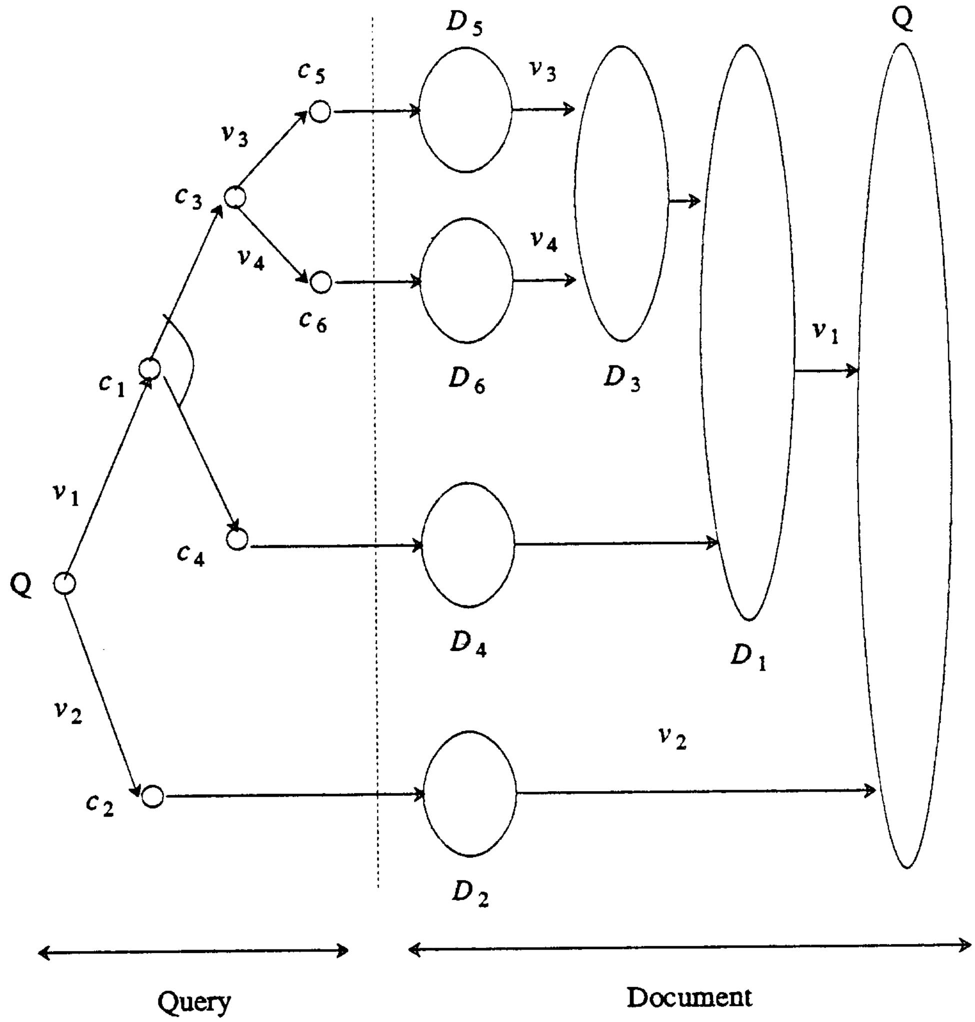


Figure 3.7. Overview of evidence aggregation.

$$D_k = D_i \cup D_j \cup D_{c_k} \tag{3.4}$$

$$m_k(X \cup D_{c_k}) = v_i \cdot m_i\ (X) + v_j \cdot m_j\ (X) \tag{3.5}$$
$$X \subset D_i \cup D_j, \quad m_i\ (X)\, m_j\ (X) \neq 0$$

where D_{c_k} stands for the set of documents indexed by the concept C_k. When X is contained only in D_i, this equation reduces to

$$m_k(X) = v_i \cdot m_i\ (X)$$

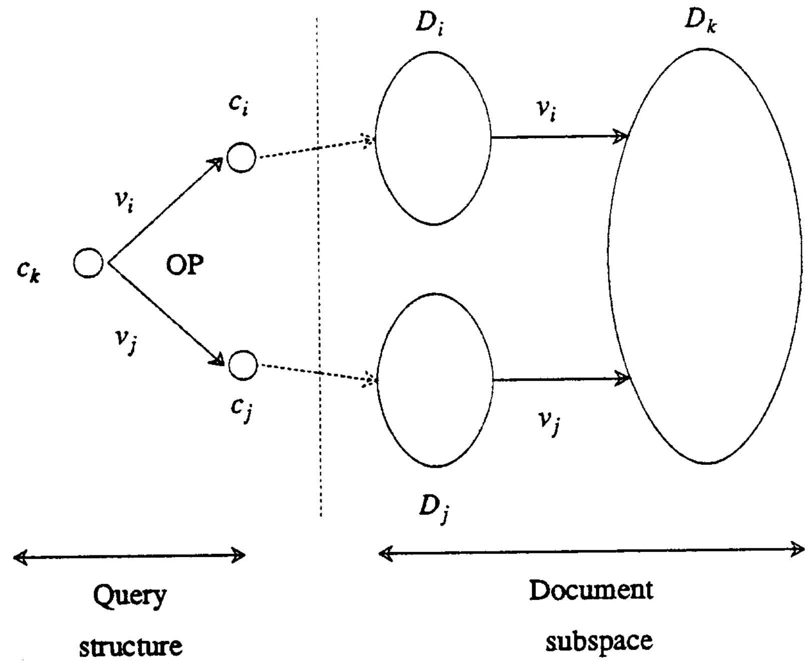


Figure 3.8. Basic operation of evidence aggregation.

because the measure of X in D_j is zero. The same arguments hold for D_j. If c_i and c_j are leaf nodes, D_i is the only subset of D_i that has a nonzero measure; actually, it is assigned to the initial value 1; this is also true for D_j. As a result, Eq. (3.6) reduces to

$$m_k(D_i) = v_i$$

$$m_k(D_j) = v_j$$

If the OP is an *AND* operation, the intersection of D_i and D_j is assigned to D_k. The derived measure of D_k is equal to the result of combining the measure of D_i and that of D_j according to Dempster's rule of combining,

$$D_k = D_i \cap D_j \tag{3.6}$$

$$m_k(X) = (m_i \oplus m_j)(X) \tag{3.7}$$

If the two measures are in conflict (D_k is equal to an empty set), the measure of X is reset to 0.

Example 3.4 (Evidence Aggregation).

Based on Examples 3.1 and Eqs. (3.4) through (3.7), the evidence aggregation process is shown in Figure 3.9.

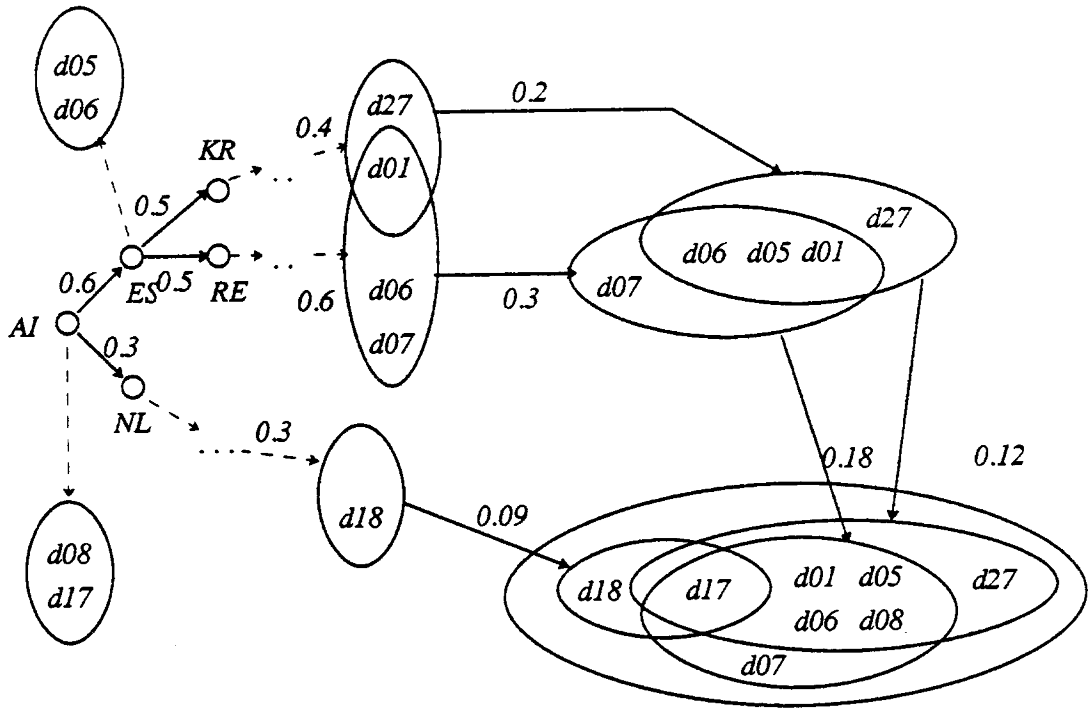


Figure 3.9. Example of evidence aggregation.

3.4. Relationship to Other Retrieval Systems

The correspondence between the new system and the Boolean model is described in this section. It is then compared to the probability model and other knowledge-based retrieval systems.

3.4.1. Correspondence to the Boolean Model

The new retrieval system subsumes the Boolean model; in other words, it can be reduced to a Boolean model and, therefore, it can produce the same result for any Boolean query. The proof is derived in Appendix A.

To reduce the new system to the Boolean model, a belief measure attached to a concept is assumed to be evenly distributed to all the associated subconcepts without any specification. Given n subconcepts associated with a given concept c_i, it is assumed that each subconcept receives an equivalent degree of belief and has the value of $1/n$ as illustrated in Figure 3.10.

3.4.2. Comparison with the Probabilistic Models

The comparison in Table 3.1 between the new system and the probability models described in Section 1.1.2 (models 1, 2 and the unified model) is analogous to that between the D-S and probability theories. It can be seen that, because the new system takes advantage of background information, it is not necessary to estimate prior probability. It also represents incomplete knowledge and encompasses information granularity. In addition, its result is a range instead of a single value. It should be noted that the comparison is made in the range of retrieval process, not the process of knowledge acquisition.

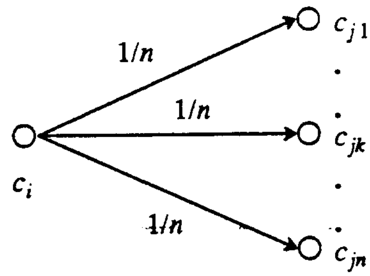
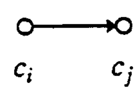
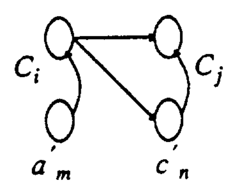


Figure 3.10. Evenly distributed belief measure.

Table 3.1. Comparison between the probabilistic models and the new system.

	Probabilistic Models (1, 2, unified)	New System
Rule		
Pairwise Conditional Independence	foreground	background
Estimation of Prior Probability	necessary	not necessary
Expression of Ignorance	no	yes
Information Granularity	no	yes
Result	single value	range

3.4.3. Comparison to RUBRIC and KADR

The new system and the other two knowledge-based retrieval systems (RUBRIC and KADR) are compared in Table 3.2. In the new retrieval system, query formulation is guided by the conceptual hierarchy based on expert knowledge, and the source of evidence is the query supplied not by the document but by the user. It is based on an extension of D-S theory that is theoretically sound. For the source of expert knowledge, it employs the experienced-based belief measure in addition to the semantic-based similarity measure.

Table 3.2. Comparison of RUBRIC, KADR, and the new model.

	RUBRIC	KADR	New Model
Query Formulation	conceptual hierarchy	Boolean structure	conceptual hierarchy guided
Evidence	document	document descriptor	query
Evidential Reasoning	related (strong evidence)	D-S theory (problems)	extended D-S theory
Expert Knowledge	semantic-based similarity		+ experience-based belief

Although all the retrieval systems attempt to solve the problem of determining the relevant documents for a given query, the problem is resolved differently by each system. Given a document as evidence, RUBRIC and KADR obtain the degree to which it is matched to a query; given a query as evidence, the new system determines the degree of belief to which a document should be selected.

4. Prototype and Implementation of the Knowledge-Based Retrieval System

This section describes the design and implementation of a prototype knowledge-based retrieval system developed for the general-purpose retrieval of information. It incorporates expert knowledge in directing the users to the information required and leads to an efficient examination of the retrieved information.

In this system, rules solicited from experts and index records from indexer are maintained in the rule base and document database, respectively. Because of its dynamics, the knowledge-based system can add/delete rules and documents at any time without interfering with the retrieval of information. The inference engine is able to derive the RSV for relevant documents.

An experiment based on this system was designed to

- demonstrate the feasibility of knowledge-based retrieval of information
- examine the response of the user and compare it to the result obtained by the system
- analyze the performance of the prototype

The participants in the experiment are potential users from the computer science community, and most of them are familiar with the fields of Artificial Intelligence. They were asked to retrieve information from a set of documents, and for each document, to judge the relevance between it and given queries.

Currently, the system is built on top of the GISTER* system [Lowrance 87]. Most of the user interface and the evidential-reasoning mechanism were adopted. The reminder of the code was implemented in Common Lisp on Symbolics 3600 and faciliates automatic query formulation and result interpretation; it cooperates with GISTER through its program interface.

4.1. Prototype

The REDER (REtrieval of Documents based on Evidential Reasoning) is a prototype of the knowledge-based retrieval system. It enables automatic query formulation and query evaluation in an evidential-reasoning environment. The final RSV for each retrieved document is based on expert knowledge and is a result of evidential reasoning.

Figure 4.1 is a block diagram of REDER. There are five major blocks in the system. The rule and document managers are responsible for the development and maintenance of the rule base and document database, respectively; they also provide an efficient method of retrieving rules and documents when requested by the other parts of the system. The index manager is a subsystem under the document manager; it accounts for the managing of index records for the set of documents and is also responsible for the communication between REDER and the indexers.

Because the function of the query manager is automatic query formulation, it interrelates with the rule manager. It also implements the process of query soliciting from the users. The major function of the performance manger is query evaluation which includes maintaining the expanded query trees and formulating the reasoning path to determine the final result. It also enables result interpretation, which displays the retrieval result in various formats, and maintains the analysis database that stores all the useful retrieval history that could be used to enhance the performance of future retrieval. The GISTER manager provides the interface between GISTER and the rest of the system. It is the core of REDER and will be discussed in the following sections.

The solid lines in this figure represent the communication between the blocks and between a manager and the remainder of a block. The dotted lines indicate the data flow in REDER; for example, the input query and rules flow into the query-manager block, and the expanded query trees are sent to the performance manager. On the other hand, related documents are transferred

* GISTER was developed by and is proprietary software of the Artificial Intelligence Center at SRI International.

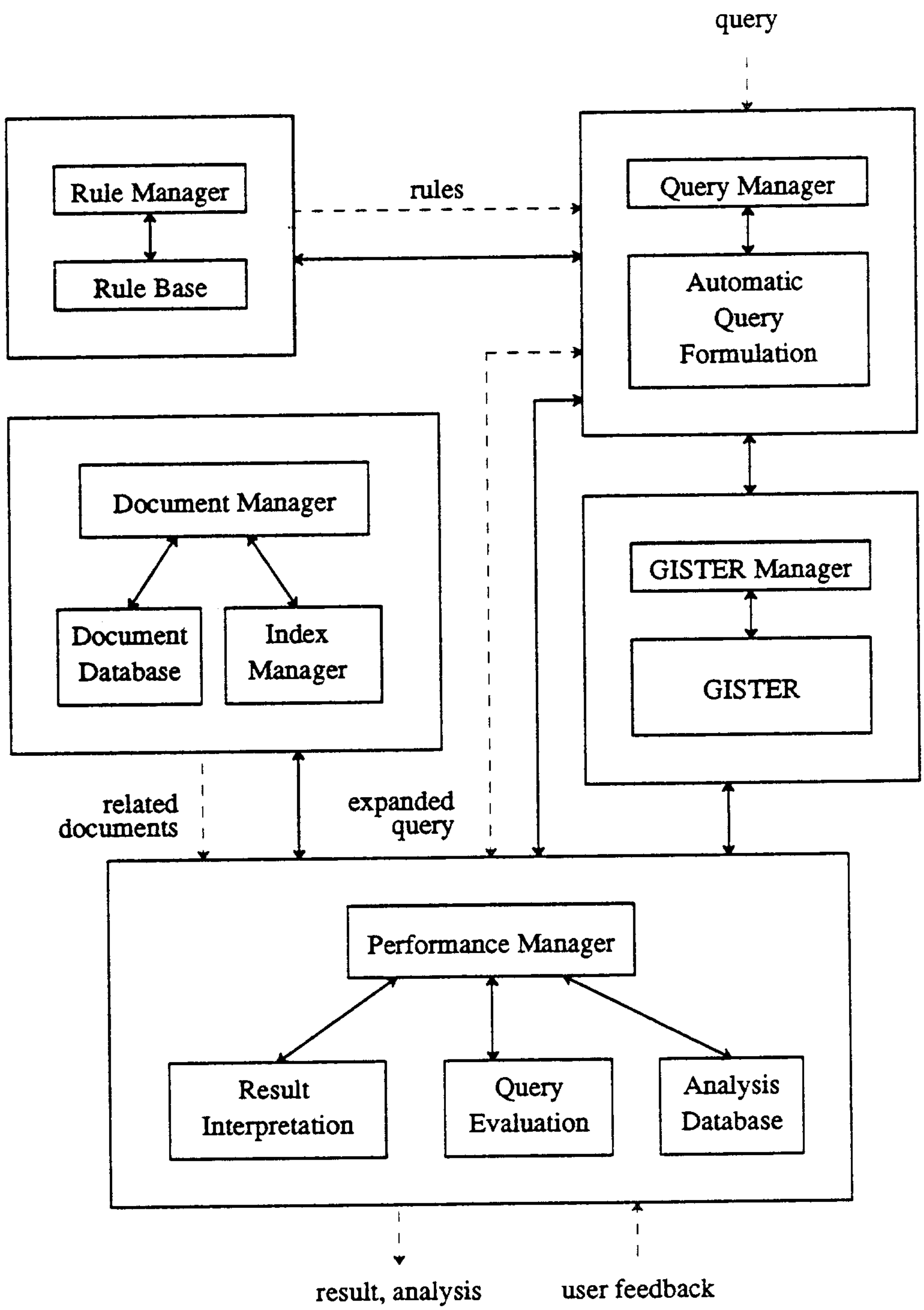


Figure 4.1. Block diagram of REDER.

from the document manager to the performance block. The result (and analysis if available) is then output from the performance manager.

4.1.1. Introduction of GISTER

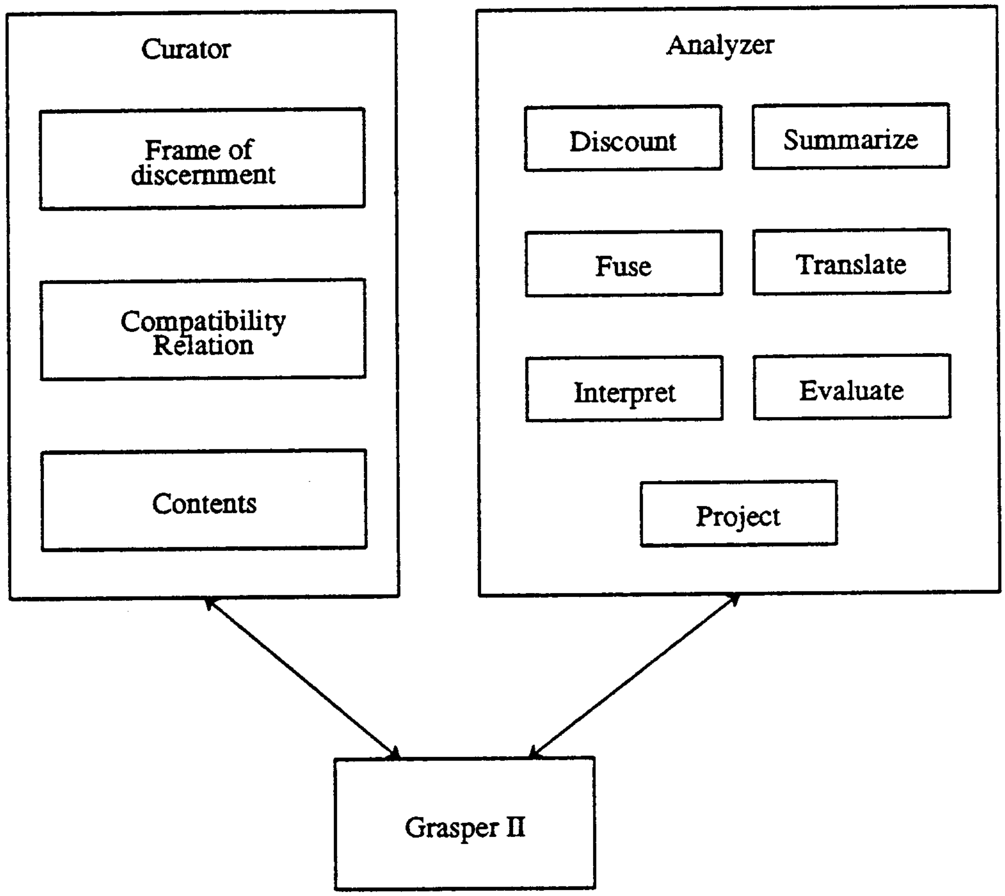


Figure 4.2. Block diagram of GISTER.

GISTER was developed as both a formal basis and a framework for implementing automated reasoning systems, especially the techniques of evidential reasoning based on D-S theory. As displayed in Figure 4.2, the major functions of GISTER [Lowrance 87] are

- to specify a set of distinct propositional spaces (frames of discernment)
- to specify the interrelationships among these propositional spaces (compatibility relationships)
- to represent bodies of evidence as belief distributions over these propositional spaces (mass distributions)
- to establish an analysis path connecting all the bodies of evidence
- to converge on spaces where the target questions can be answered

The analysis function is detailed by the following operations:

- discounting -- to assign credibility to the source of a body of evidence

- fusion -- to combine different bodies of evidence
- translation -- to translate the influence of a body of evidence from one context to a related one
- evaluation -- to trigger a series of evidential-reasoning activity
- summarization -- to summarize the currently available body of information
- interpretation -- to determine the final measurement of a specified statement based on a given body of evidence
- projection -- to project a body of evidence to a temporally related one

In GISTER, a set of frames of discernment and the compatibility relationship among them form a *gallery*. As illustrated in Figure 4.2, the Curator is the manager of galleries and is responsible for the development, update, and maintenance of the elements of a gallery. The Analyzer contains the above operations, and Grasper II is the manager of the graphic interface on which GISTER is built.

4.1.1.1. User Interface

GISTER also provides an interactive, menu-driven, graphic interface that enables the users to understand the reasoning process through graphic structures. Following the reasoning path is simplified, and attention is focused on certain parts of the path.

A graphical symbol is assigned to each basic element in a gallery as illustrated in Figure 4.3. Table 4.1 summarizes some of the functions of GISTER.

It is also possible to use the Lisp program to communicate with GISTER through the programmer interface and the Lisp window provided by GISTER. Basically, all of the operations supported by user inference are included in this interface. They are called procedures and are specialized Lisp functions designed for GISTER. When invoked, they can access all of the GISTER data structures; however, the internal procedures do not synchronize with the GISTER display. Direct access to the Grasper is necessary to update the display.

Following are examples of the procedures defined in the programmer interface. The procedure names are in capital letters, and arguments are in brackets. The procedure names ending with a question mark indicate predicate procedures.

- CREATE-ANALYSIS-SPACE [analysis gallery]
- CREATE-FUSION-RELATIONSHIP
  [evidence-nodes analysis &OPTIONAL location]
- EVALUATE-INTERPRETATION-RELATIONSHIP
  [interpretation-node analysis]
- PROJECTION-NODE? [node space]
- INTERPRET [massfun propositions bias fod]
- CREATE-RELATIONSHIP-EDGE [element1 element2 relation gallery]
- TRANSLATION-RELATION? [possible-translation gallery]

4.1.1.2. Example

This section illustrates GISTER by an example borrowed from Lowrance [Lowrance 87]. The task is to locate a ship at a specific time through several reports regarding its recent locations

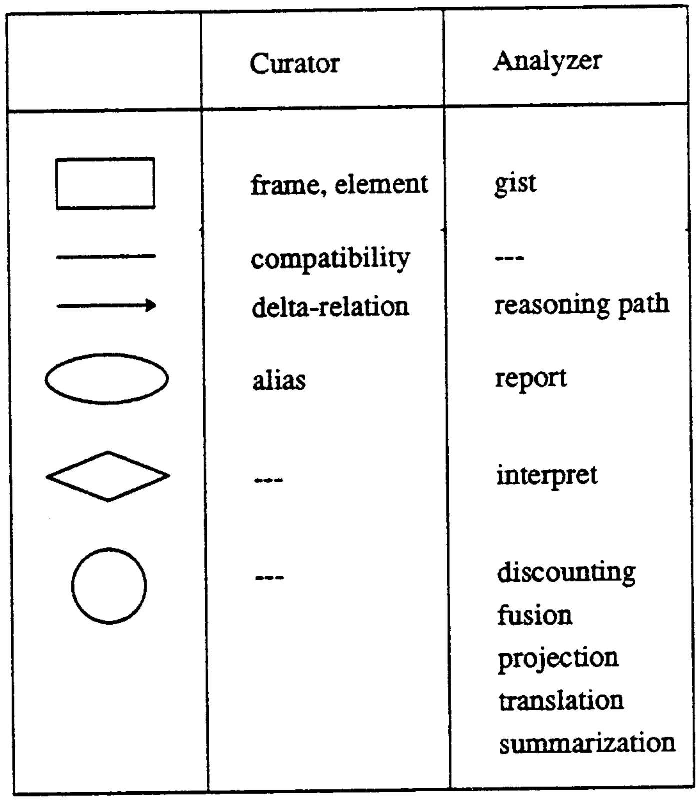


Figure 4.3. Graphic symbols in GISTER.

Table 4.1. Graphic functions of GISTER

Function	Description
create	create gallery, analysis, contents, report, ...
backup	save the current status of a graph
revert	restore the previous saved status
examine	examine contents of analysis nodes
ancestors	highlight ancestors nodes in a reasoning path
descendant	highlight descendant
print-draw	print or draw a graph to file, paper, or window

and activities. The knowledge base of this problem consisting of knowledge concerning possible locations and activities and the relationship between them and the reasoning procedure based on the available reports.

A gallery in GISTER signifying the knowledge problem, contains two frames (locations and activities) and a locations-activities relationship. The location frame (Figure 4.4) contains several elements -- *channel, refueling-dock, loading-dock, zone1, zone2,* and *zone3*. *At-sea, in-*

port, and *docked* are aliases defined on subsets of the elements. The compatibility relationship between the two frames, illustrated in Figure 4.5, links the locations to the possible activities that could occur within these locations, such as *enroute, tug-escort, refueling, loading*, and *unloading*.

The *ship* gallery contains two additional relationships, the so-called *delta-relationships* that map a frame to itself to reflect the possible change over the time frame within which the reports arrive, as illustrated in Figure 4.6. For example, if the ship is reported to be in a *tug-escort* activity in report1 and if report2 is one time-slice after report1, the activity of the ship when report2 arrives could be *enroute, refueling*, or *unloading*. A similar relationship is also defined for the delta-locations.

The second problem is the formulation of an analysis (reasoning process) based on the available body of evidence. This body of evidence contains three reports. Report1 confirms that the ship was at channel with 70 percent chance and at zone1 with 30 percent. Report2 follows and states that the ship was in-port, and report3 is the last to arrive and indicates that the ship was loading.

The first subgoal in the analysis task is to project all the reports onto a common time slice and to fuse them at that point. The second is to direct the reasoning to the location frame in which the entire problem is addressed.

Figure 4.7 describes the entire process. The results reveal a strong belief that the ship is *in-port*, with [Bel, Pls] pairs equal to [0.9, 1.0] and that it could be *docked* [0.79, 0.85]. In the subwindow, the numbers before the bar chart indicate the center point of the range of the belief measure, based on the assumption that bias does not exist throughout the reasoning process.

Figures 4.4 through 4.7 are listed in the following pages.

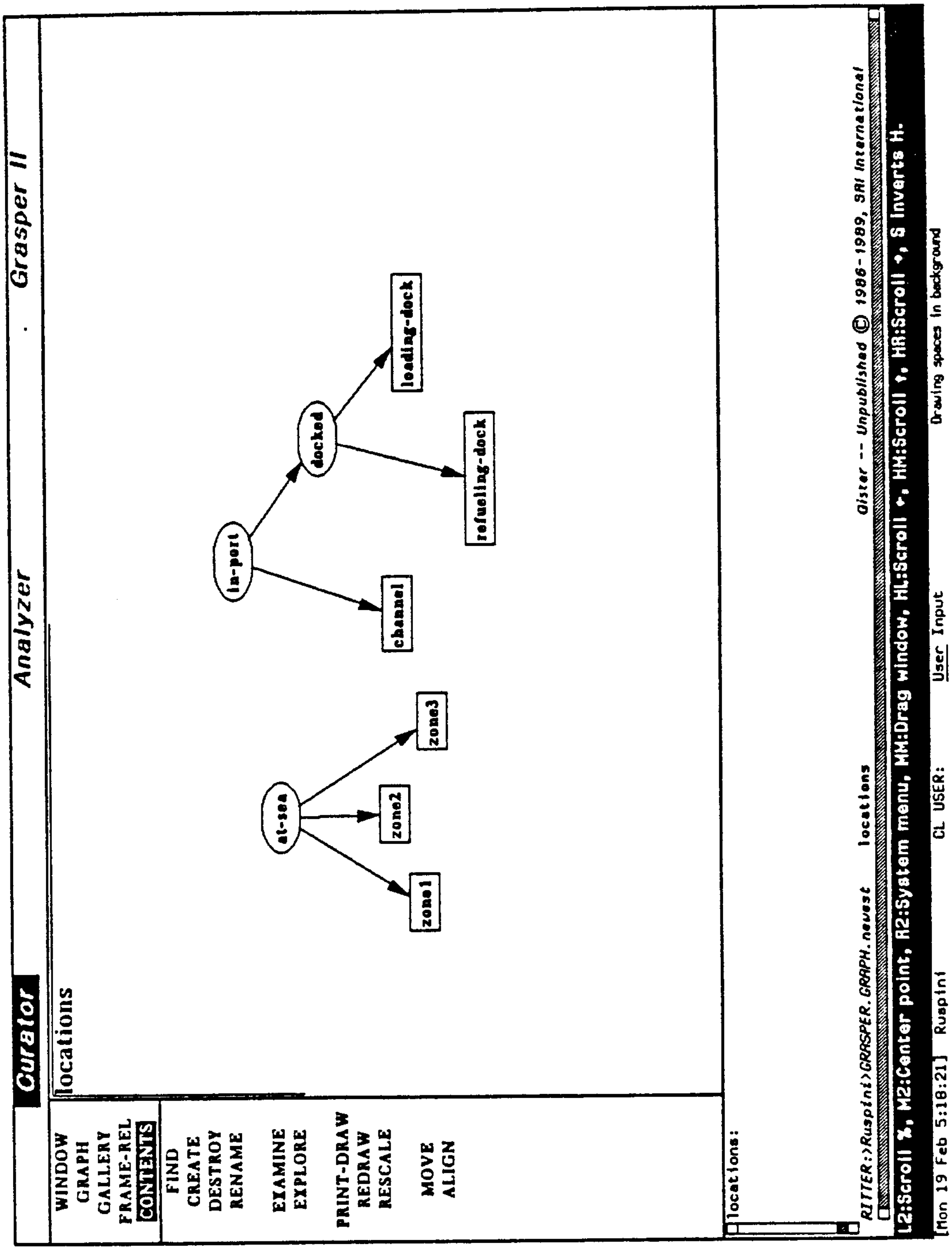


Figure 4.4. Location frame of the ship gallery.

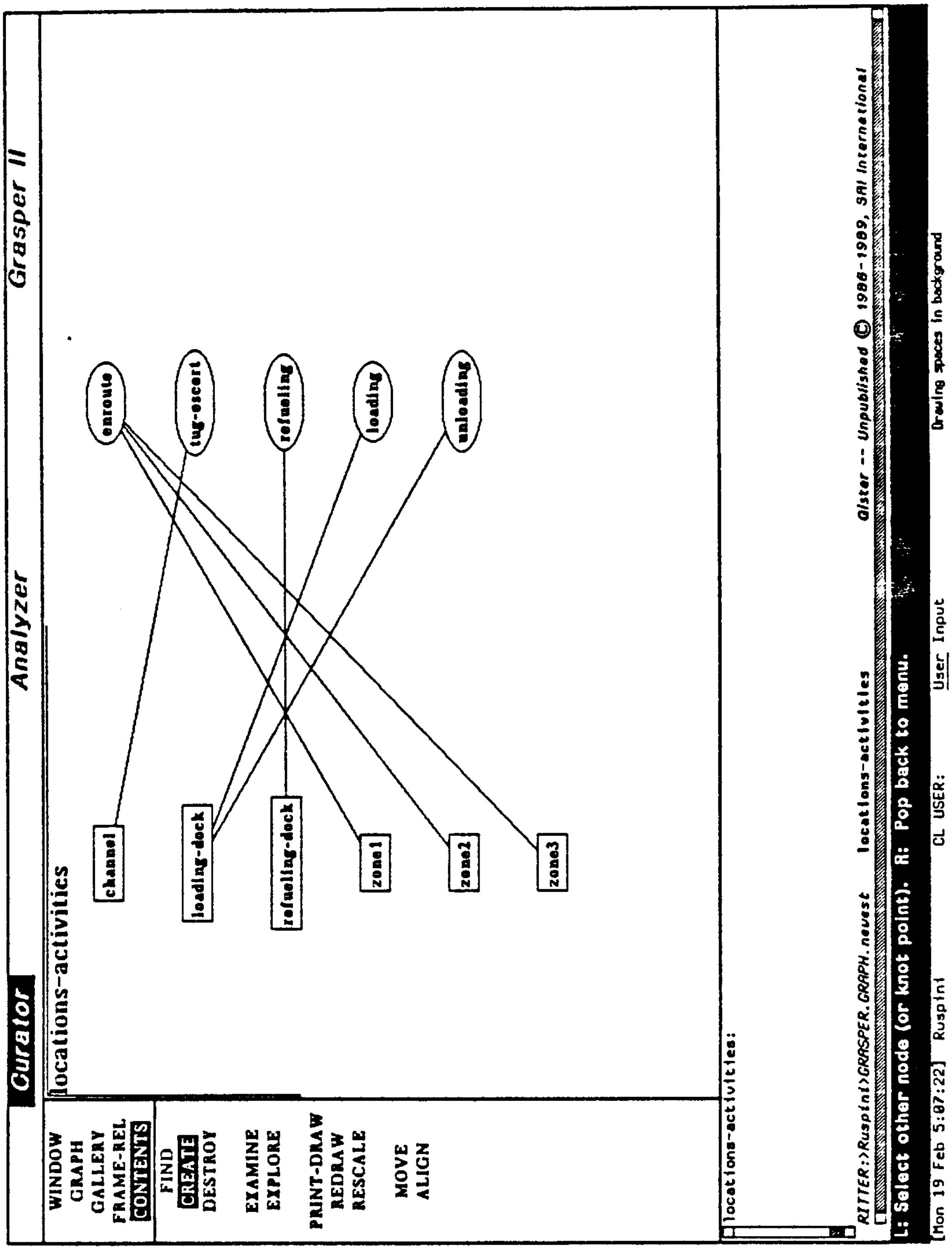


Figure 4.5. Locations-activities relation.

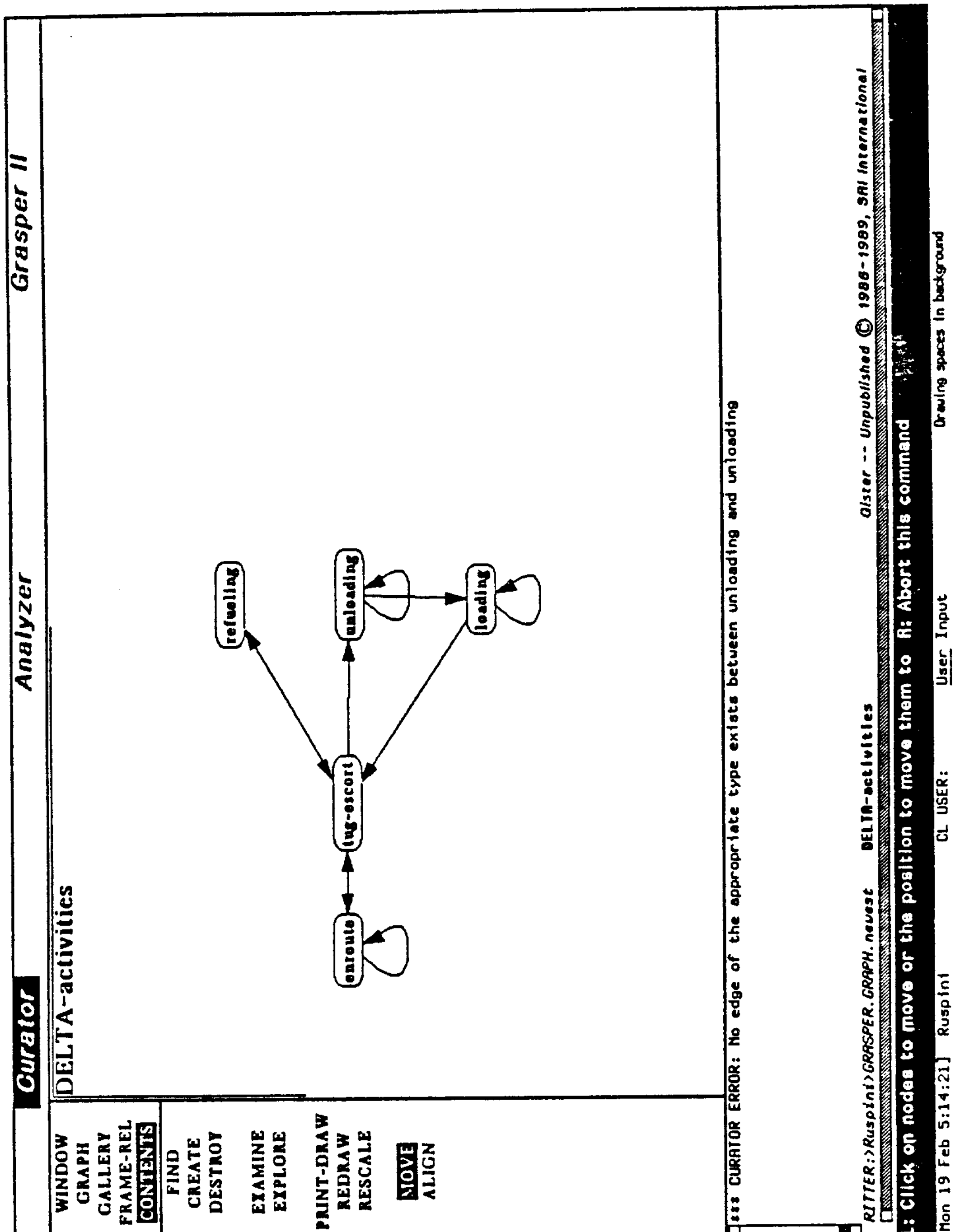


Figure 4.6. Delta-activities relation.

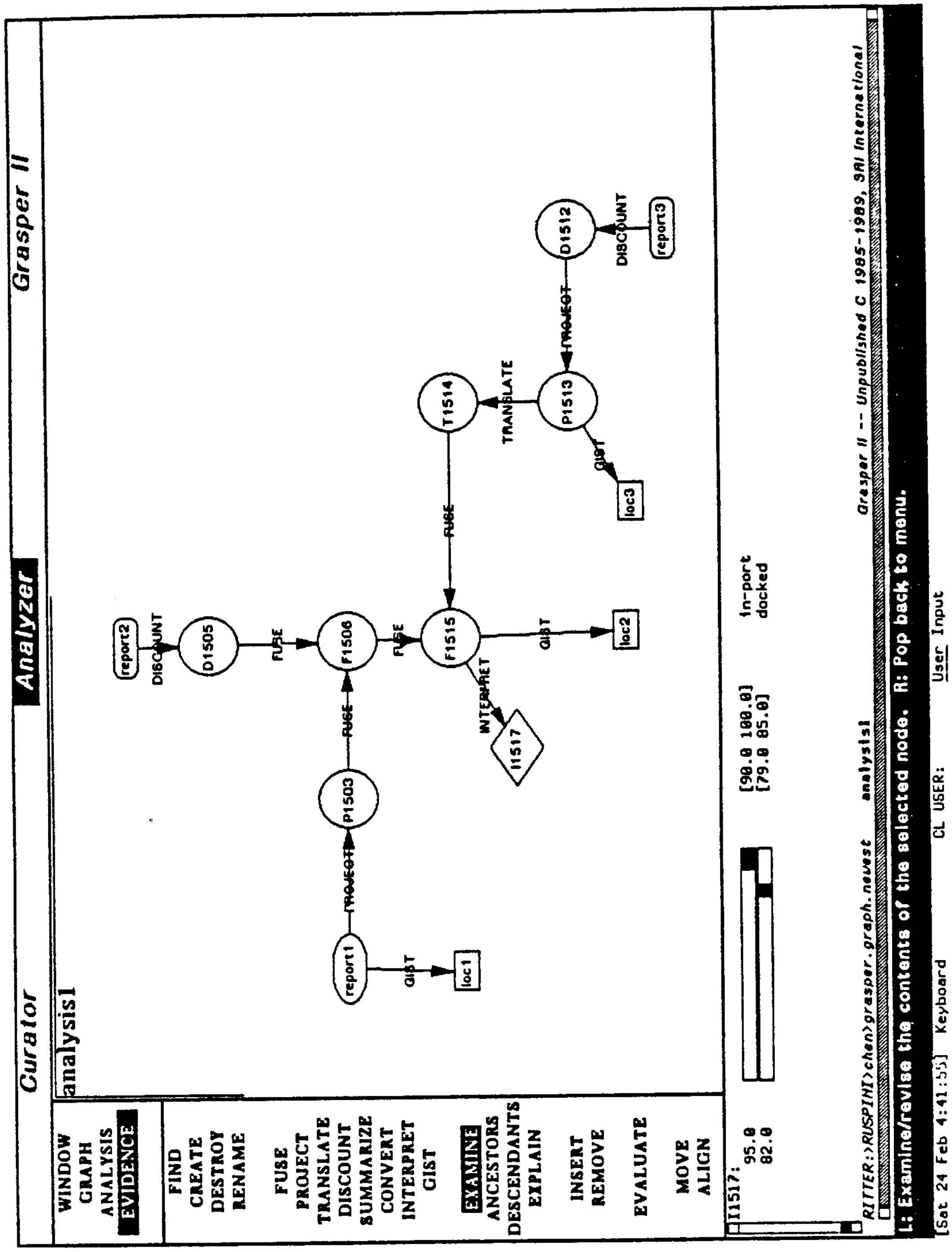


Figure 4.7. Analysis of the ship-location problem.

4.1.2. Beyond GISTER

This section describes the main procedure of REDER and functions that are not supported by GISTER. The code is in a pseudo-code format that closely resembles the Common Lisp but does not reflect all the details of the real code.

4.1.2.1. Main Procedure

```
(defun REDER ()
 (init)
 (loop-forever
  (setq Query-tree (QUERY-MANAGER))
  (setq Result
   (PERFORMANCE-MANAGER
    '(query-evaluation Query-tree)))
  (cond ((analysis-necessary)
       (PERFORMANCE-MANAGER
       '(analysis Result))))))

(defun QUERY-MANAGER ()
 (init)
 (setq Query (query-soliciting))
 (setq Query-tree (automatic-query-formulation Query))
 (return Query-tree))

(defun RULE-MANAGER (message)
 (init)
 (case message
  ('insert (create-rule Rule-base))
  ('delete (delete-rule Rule-base))
  ('update (modify-rule Rule-base))
  ('compile (compile-rule Rule-base))))

(defun DOCUMENT-MANAGER (message)
 (case message
  ('insert (add-new-document Document-database))
  ('delete (remove-document Document-database))
  ('update (modify-document Document-database))
  ('invert-index (invert-index Document-database Index-database))))

(defun PERFORMANCE-MANAGER (message)
 (case message
  ('query-evaluation
   (setq Report
    (report-interpretation-formulation Query-tree))
   (result-interpretation Report))
  ('analysis
   (setq Feedback
    (feedback-soliciting Report))
   (analysis Report Feedback Analysis-database))))
```

4.1.2.2. Recursive Report-Interpretation Formulation

Although GISTER provides a sound environment for the basic operations of evidential reasoning, it includes only those functions based on the original Dempster-Shafer theory. It does not support the uncertainty operations defined in the extended theory.

First, it does not address the conditional belief measure attached to the link of a rule. In GISTER, the link in the reasoning path has a default belief measure equal to 1 which reflects the unconditional compatibility relationship among frames of discernment.

Second, it rejects the chaining operation based on the conditional belief measure. In GISTER, the chaining operation relies on the translation/projection relations based on the set-inclusion relationship.

Third, it does not support the basic operation designed for knowledge-based retrieval in REDER. In GISTER, the discounting operation is useful when discounting an existing measure; however, it must function on an entire frame of discernment and cannot be applied to only a part of a measure. In addition, all the reports (sources of evidence) can be combined only through the fusion operation; two partial reports cannot be combined into a new report.

To solve these problems, a recursive report-interpretation formulation was designed. This procedure, known as RI-formulation, takes a query tree and generates a report and interpretation recursively at each level of the tree. After a report is produced, GISTER performs the basic operations and the report is returned for later use at a higher level. At the end, a final report is formulated at the top level and interpretation yields the retrieval result. Following is the pseudo-code of this algorithm.

```
(defun RI-formulation (Query-tree)
 (init)
 (cond ((leaf-node? (child-of Query-tree))
      (loop-init)
       (loop foreach Child of Query-tree
           (setq report
                 (append-to-report report
                  (make-report (get-link-measure Child)))))
       (make-interpretation Query-tree report)
       (make-gist Query-tree report)
       (return report))
      (t
       (loop-init)
       (loop foreach Child of Query-tree
         (cond ((is-OR-op? (get-op Query-tree))
                (setq temp-report
                      (RI-formulation Child))
                (setq discount-report
                      (discount-report temp-report
                        (get-link-measure Child)))
                (cond ((is-exist-in? temp-report report)
                       (setq report
                         (update-report report
                           discount-report)))
                      (t
                       (setq report
                         (append-to-report report
```

```
                discount-report)))))
        (t
         (setq report
               (fuse-report temp-report report)))))
    (make-interpretation Query-tree report)
    (make-gist Query-tree report)
    (return report))))
```

4.2. Experiment Using the Knowledge-Based Retrieval System

This section describes an experiment developed for the REDER system. Its design is first discussed, and the results serve as a basis for the performance evaluation of REDER.

4.2.1. Design

The major task of this experiment was to ask a group of potential users to perform an actual retrieval from a set of documents and to compare the results to those obtained by REDER. For this comparison, a set of queries was selected. The users were asked to assume that these queries reflected their information needs and to answer questions regarding the relationship between them and the documents. They were also asked to study all the documents in the document set.

The participants of this experiment were volunteers with computer science or electrical engineering backgrounds. Many of them were Ph.D. candidates at the University of California, Berkeley, and some were visiting scholars. All were familiar with artificial intelligence. Because the experiment was conducted on an individual basis, the answers from one user could not affect those from another.

4.2.1.1. Problem Domain

Artificial intelligence was selected as the problem domain, and AIList was chosen as the source of documents. AIlist is a message service across the computer network and is a common place where people interested in AI could share their knowledge. Because of the diversity of topics in this field, however, the receiver of the AIList may not be interested in all of the message. In addition, although the message has been edited, its volume is still very large, and reading all of it is time consuming.

This problem can be solved by knowledge-based retrieval of these documents. On one hand, expert knowledge could direct the user to the documents of interest; on the other hand, the ranked list of retrieved documents is a helpful guide to determine which documents to examine first.

The following queries were selected for the experiment. The users were asked to assume that they were their information needs and to judge their relevance to the set of documents:

- Query 1. artificial-intelligence
- Query 2. expert-system
- Query 3. natural-language
- Query 4. expert-system OR logic-programming
- Query 5. expert-system AND logic-programming

The first three queries are single concepts formed as general terms in the field of AI. The last two are Boolean queries that combine two general terms. The purpose of this choice of queries was to confirm the capability of REDER to expand them into more specific terms and to retrieve relevant documents at the same time.

4.2.1.2. Rule Base

The following list of rules was developed for this experiment. Each rule maps a concept into a set of more specific terms. The number in parentheses indicates the conditional belief measure obtained from the expert; its default value is 1.

AI -> Natural-language (0.3), Expert-system (0.5)
    Logic-programming (0.2).
Natural-language -> Machine-translation (0.15),
    Electronic-dictionary (0.3), Knowledge-representation (0.5).
Expert-system -> Knowledge-representation, Reasoning
Reasoning -> Symbolic-reasoning (0.3), Numerical-reasoning (0.7)
Numerical-reasoning -> Certainty-factor (0.1),
    Numerical-reasoning-theory(0.9).
Numerical-reasoning-theory -> (Approximate-reasoning,
    Plausible-reasoning, Causal-network)

4.2.1.3. Document Sets

Although the large volume of documents is not a problem for knowledge-based retrieval, it is not practical in this experiment because the user is asked to examine every document in the document set so as to judge the ability of the system to retrieve relevant and, at the same time, to reject nonrelevant documents. It is more important, therefore, for the user to study a small volume of documents rather than to read only a part of a larger volume.

As a result, 30 documents were selected from the AIList, each spanning a time frame over the past six months and chosen from different areas in the AI field. Following is a list and a brief description of each document.

d01 Adding Forward Chaining and Truth Maintenance to Prolog
d02 Text Analysis for Text Retrieval
d03 Generating Plausible Diagnostic Hypotheses with
    Self-processing Causal Networks
d04 Concurrent Engineering Design
d05 The Importance of Expert Systems
d06 Expert Systems Minitrack
d07 "The Four References" -- A Study of Mind
d08 Call for IJCAI-89 Student Volunteers
d09 The Third Annual Workshop on Blackboard Systems
d10 Rule Generation for Plausible Inference
d11 Paradoxes of Indirect Discourse
d12 Call for Papers -- Symposium on Principles of Database Systems
d13 Welcome to the Task Force on Expert Systems
d14 Dictionaries in the Electronic Age
d15 Bootstrapping One-sided Learning
d16 Pundit's First French Lesson:
    The PRATTFALL Machine-Translation Module

d17 Nobel Laureate Herbert Simon to Visit Rutgers
d18 Logic Programming: A Wrong Road for AI
d19 Guaranteeing Serializable Results in Parallel Production Systems
d20 Parallel Processing and AI
d21 Workshop on Term Subsumption Languages in
Knowledge Representation
d22 The Matrix of Biological Knowledge
d23 On the Proper Place of Connectionism in Modeling Our
Behavioral Capacities
d24 Computational Value Analysis
d25 Fodor's Perverse Frame Problem and Its Implications for
Scientific A.I.
d26 Data Communications Job Offered
d27 Has Representation Been Naturalized?
d28 Automating Software Design
d29 Systems Architects Job Offered
d30 Software Engineers Needed

An index record assigned to each document contained keywords that addressed the document. An inverted index could be created automatically to cover the entire set of documents. Following is a list of all the indexes.

d01 prolog, reasoning, knowledge-representation
d02 text-analysis, text-retrieval, natural-language, plausible-reasoning,
lexical-analysis, knowledge-representation
d03 connectionism, causal-network
d04 concurrent-engineering, knowledge-representation
d05 expert-system
d06 expert-system, knowledge-base, decision-support, knowledge-representation,
reasoning
d07 reasoning
d08 artificial-intelligence
d09 blackboard-system
d10 knowledge-base, plausible-reasoning, deductive-reasoning
d11 natural-language
d12 database, knowledge-base
d13 expert-system
d14 electronic-dictionary, expert-system
d15 learning-algorithm
d16 machine-translation
d17 artificial-intelligence
d18 logic-programming, prolog
d19 parallel-production-system
d20 parallel processing
d21 knowledge-representation
d22 knowledge-representation, knowledge-base
d23 connectionism, neural-network
d24 decision-analysis
d25 frame-problem

d26 data-communication
d27 knowledge-representation
d28 automatic-software-design
d29 network-management
d30 software-engineer

The users were asked to judge the relevance of each document to the set of queries selected plus optional questions that ~~would help to analyze the results~~.

4.2.1.4. Problem Formulation in REDER

The retrieval experiment was formulated in REDER, and the displays provided by GISTER are presented in Figures 4.8 through 4.11. Figure 4.8 is the content of a document frame for the first retrieval (artificial-intelligence). All the documents are elements of the frame, and several aliases are defined and later used by the reports. Figure 4.9 is the analysis at the leaf-node level, showing the report generated by REDER for the sets of documents associated with the concepts *natural-language*, *logic-programming*, and *numerical-reasoning*. Figure 4.10 is the analysis at the top level where the recursive reasoning path has converged into a single report. A section of the final result obtained by examining the interpretation is displayed in the Lisp subwindow.

The formulations of the other retrievals are similar to the formulation displayed here, but they are at different levels. The *OR* relation in the fourth retrieval is represented by an *OR* node in the document space, and the conditional belief is equally divided by its child node. The *AND* relation in the last retrieval is formulated by the fusion operation and is displayed in Figure 4.11.

Figures 4.8 through 4.11 are displayed in the following pages.

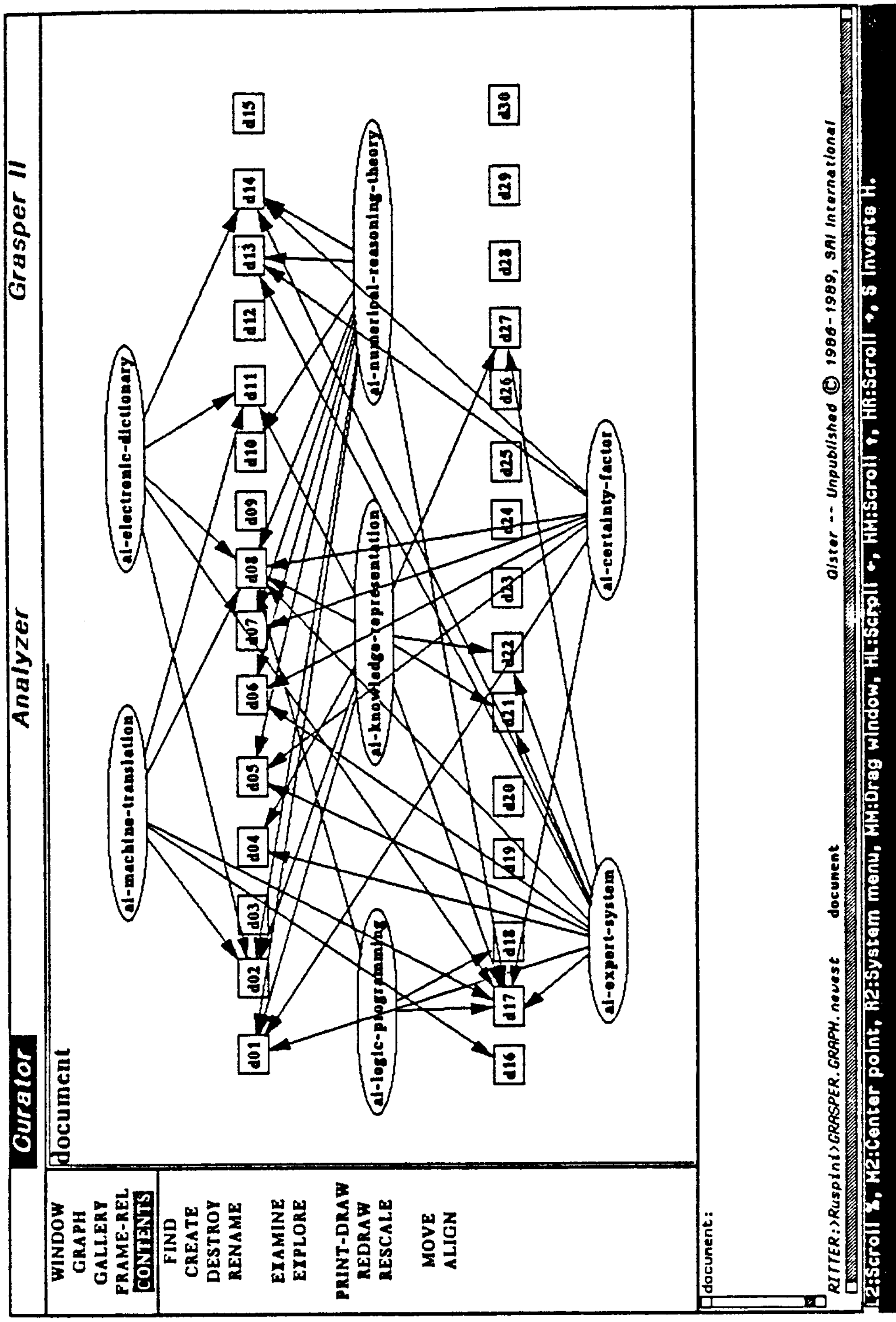


Figure 4.8. Document frame for the retrieval experiment.

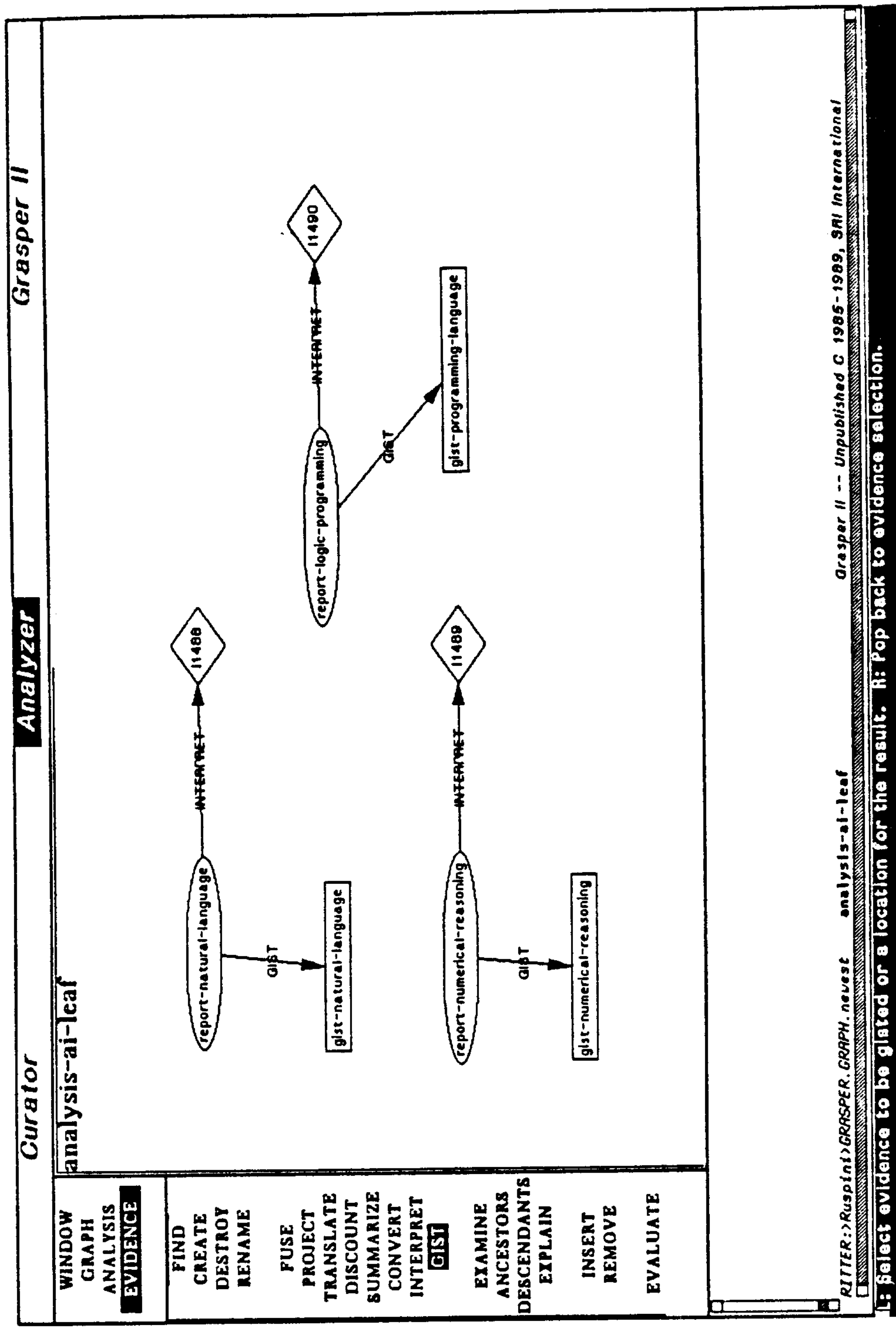


Figure 4.9. Analysis at the leaf-node level.

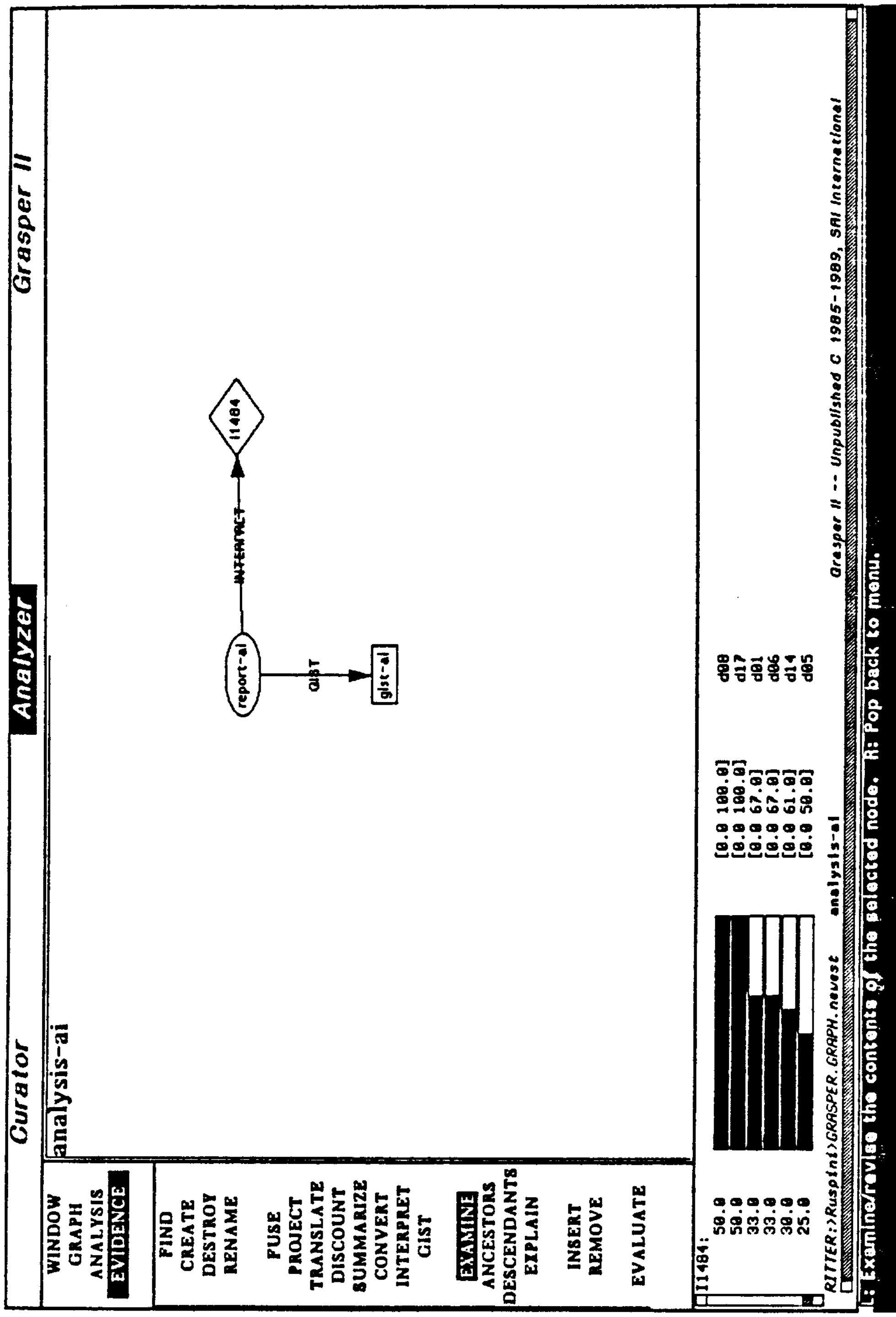


Figure 4.10. Analysis at the top level.

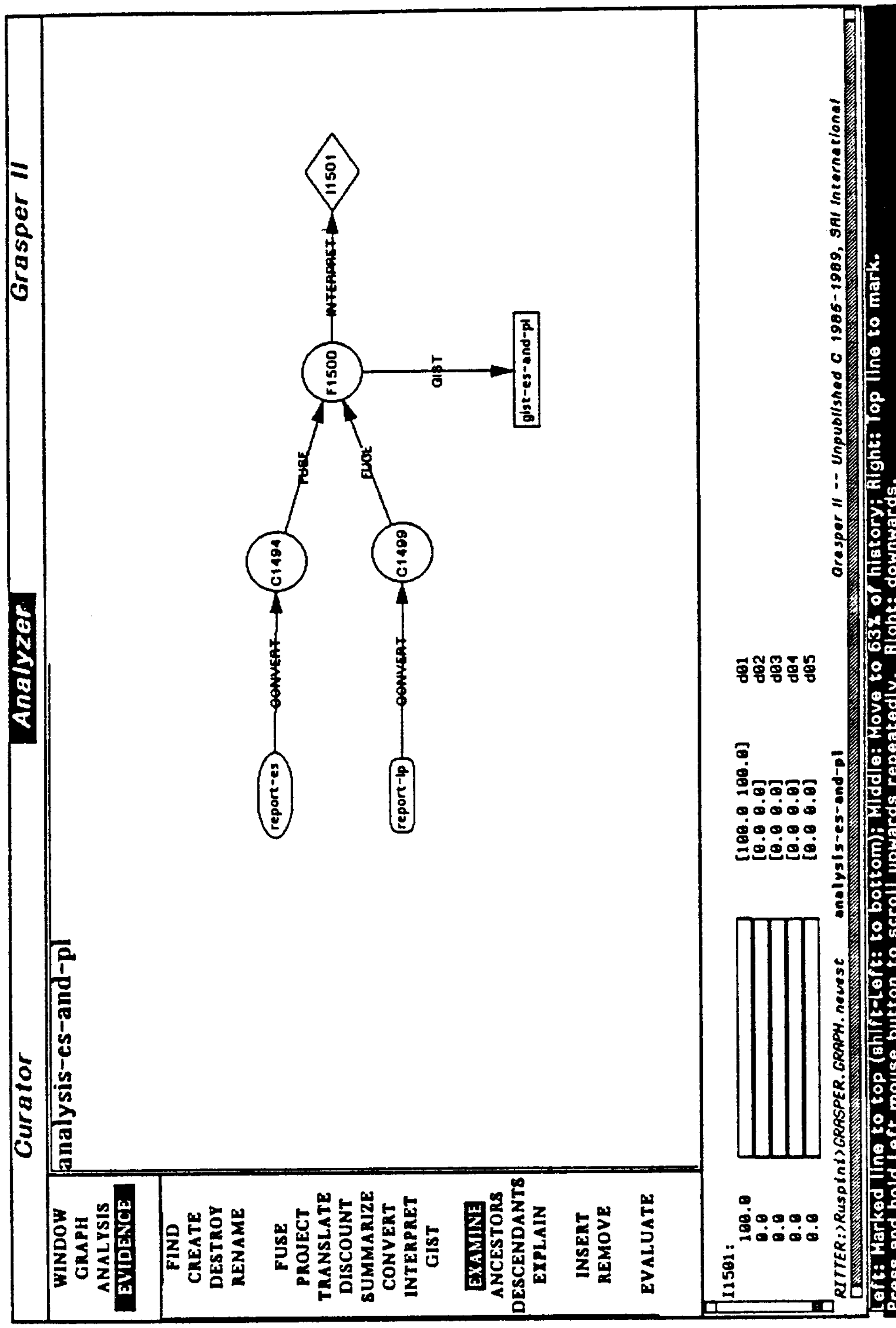


Figure 4.11. Fusion operation for the AND query.

4.2.2. Results

The experiment results are divided into those obtained by REDER and those produced by the users. They are listed in separate tables and are compared via the performance measure in Section 4.3.

4.2.2.1. Obtained by REDER

The results generated by REDER are listed in Table 4.2, where di denotes the documents and qj represents the queries. ~~These are the results of formulating each query~~ in REDER and obtaining the RSV value for each document.

Table 4.2. Result obtained by REDER.

Document	q1	q2	q3	q4	q5
d01	0.67	1	0.55	0.5	1
d02	0.46	0.32	1	0.16	
d03	0.16				
d04	0.42	0.5	0.55	0.25	
d05	0.5	1		0.5	
d06	0.67	1	0.55	0.5	
d07	0.25	0.5		0.25	
d08	1			0.5	
d09	0.42				
d10	0.16	0.32		0.16	
d11	0.3		1		
d12	0.42				
d13	0.5	1		0.5	
d14	0.61	1	0.35	0.5	
d15	1				
d16	0.06		0.2		
d17	1			0.5	
d18	0.2			0.5	
d19	0.5				
d20	1				
d21	0.42	0.5	0.55	0.25	
d22	0.42	0.5	0.55	0.25	
d23	1				
d24	1				
d25	1				
d26					
d27	0.42	0.5	0.55		
d28	0.5				
d29					
d30					

4.2.2.2. Obtained by Users

User responses are listed in Tables 4.3 through 4.7. Each table lists the results from a single query. Twelve users participated. Although assignment of a degree of relevance beyond a simple yes/no answer was optional, most of the users were able to provide this measure.

Table 4.3. User response for retrieval 1
Query: Artificial-Intelligence

Document	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12
d01	0.1	1	0.8	1	1	1	0.8	0.8	1	0.8	1	1
d02	0.1	1	0.9	1	1	1	0.8	0.6	1	0.3	0.9	1
d03	0.2	0.5	0.5	1	1	1	0.6	0.7	1	0.5	1	1
d04	0.1	0.2	0.2	0.3		0.1	0.4	0.2	0.2		0.4	1
d05	0.2	0.8	0.8	1	1	1	0.6	0.9	1	0.3	1	1
d06	0.2	0.5	0.8	1	1	1	0.6	1	1	0.7	1	1
d07	0.1	0.2	0.7	0	0	1	1	0.7	0	0.5	1	1
d08	0.1	0.6	0.7	0	1	0.1	0.3	0.8	0.3	0	0.1	1
d09	0.8	0.9	0.8	0.5	1	1	0.8	0.6	1	0.8	0.2	1
d10	0.5	1	0.9	1	1	1	1	0.9	1	0.8	0.9	1
d11	0.4	0.6	0.8	0	1	1	0.8	0.7	1	0.9	1	1
d12		0.4	0.6	1	1	1		1	0.2	0.5	0.5	1
d13		0.5	0.8		1		0.7	0.9	1	0.3	0.5	1
d14			0.4	1		0.8				0.1	0.2	1
d15	0.4	0.6	0.6	1	1	1		0.8	1	0.5	0.1	1
d16	0.1	0.9	0.7	1	1	1	0.8	0.8	1	0.5	1	1
d17	0.1	0.4	0.3	1		1	1	0.8	1	0.2	0.05	1
d18	0.4	1	0.7	1	1	1	0.8	0.8	1	0.8	1	1
d19		0.8	0.8	1	1	1	0.8	0.3	1	0.4	1	1
d20	0.6	0.9	1	1	1	0.5	1	1	1	0.5	0.5	
d21	0.2	0.7	0.6	1	1	1	0.8	0.3	1	0.5	0.2	1
d22	0.1	0.3	0.4		1	1	1	0.3		0.3	0.5	1
d23	0.1	0.3	0.7	0.2	1	1	0.6	0.8	1	0.6	0.7	
d24	0.5	0.7	0.5	1	1	0.9	1	0.8	1	0.5	1	
d25	0.5	0.7	1	0.5	1	1	0.8	0.9		0.9	0.9	1
d26												
d27			0.6			0.5	0.4		0.2		0.3	
d28	0.1	0.3	0.6		1	1	0.4	0.4	1	0.2	0.1	
d29			0.2									
d30	0.05	0.2	0.6	0.1		0.1	0.2	0.4	1	0.5	0.1	

4.3. Performance Analysis

It is difficult to evaluate a retrieval system because there is still no consensus among the researchers in this field as to which method should be used for all the retrieval systems [Rijsbergen 79]. The following categories of *relatedness*, however, help to determine the value of a system to the users [Buckland 83]:

Table 4.4. User response for retrieval 2.
Query: Expert-System

Document	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12
d01	0.2	0.5	0.3	1		0.2		0.7		0.5	1	1
d02	0.1		0.2					0.6			0.9	1
d03				1			0.5		1			
d04	0.1	0.2	0.2	0.3			0.6				0.2	1
d05	0.9	1	1	1	1	1	1	1	1	0.9	1	1
d06	0.9	1	1	1	1	1	1	1	1	1	0.9	1
d07	0.1						0.4					1
d08												
d09	0.8	0.8		0.2		0.3	0.3			0.6	0.2	
d10	0.4	0.9	0.9		1	0.8	0.2		1	0.5	0.6	1
d11			0.2			0.3				0.5		1
d12			0.9	0.3	1	0.5	0.6	0.1	0.2	0.5	0.3	1
d13	0.2	1	1	1	1	1	1	1	1	0.6	0.5	1
d14	0.2		0.6			0.5			1		0.2	1
d15		0.3	0.2					0.8				
d16		0.9	0.4				0.4	0.6	1			
d17							0.3					
d18			0.2	1		0.2				0.4		
d19	0.9	0.6	1	1	1	0.8	1	0.5	1	0.6	0.9	
d20			0.5									
d21		0.8	0.8		1	0.5	0.4			0.2		1
d22	0.1		0.3				0.8	0.4	1			1
d23												
d24				1							0.8	
d25			0.6									
d26												1
d27			0.4									
d28	0.2	0.5	0.7		1	1		0.3	1	0.6		
d29												
d30												

- Relatedness I: responsiveness -- the relatedness of the data retrieved to the inquiry as posed in terms of the attributes used as the basis for retrieval
- Relatedness II: pertinence -- the relatedness between the properties of data with respect to the attributes used for retrieval when these properties are related but not identical
- Relatedness III: beneficiality -- the beneficial effects of using the system; social values are implied

The measurement of the last two categories is indirect, time-consuming, and automatically difficult; heavy involvement of the users is required.

Table 4.5. User response for retrieval 3.
Query: Natural-Language

Document	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12
d01							0.8					
d02	0.5	1		1	1	1	0.8	0.9	1	0.7		1
d03												
d04												
d05												
d06			0.2	1		0.1		0.3	0.2	0.2	0.6	
d07	0.1					0.5	0.7		1	0.3	0.4	
d08												
d09												
d10												
d11	0.9	0.4	0.7	1	1	0.5		0.3	1	0.8	0.6	
d12								0.1				
d13												
d14	0.5	0.4			0.3	0.6		0.8	1	0.4		
d15						0.1						
d16	0.9	1	0.8	1	1	1	1	1	1	1	0.9	1
d17												
d18												
d19												
d20			0.2									
d21			0.3				0.7	0.8			0.2	1
d22												1
d23												
d24												
d25												
d26												1
d27												
d28												
d29												
d30												

The knowledge-based retrieval system in this research used relatedness I in selecting relevant documents; however, the users were encouraged to choose relatedness II to determine the relevance between queries and documents. The performance analysis of the system is based on the judgement of the user.

In addition to the problem of defining relatedness, it is necessary to predict the need for other measurements that will reflect the ability of the system to satisfy the user. The following six measurable quantities were proposed [Cleverdon, Mills and Keen 66]:

- *coverage* of the collection (extent to which the system includes relevant matters)

Table 4.6. User response for retrieval 4.
Query: Expert-System OR Logic-Programming

Document	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12
d01	1	1	0.5	1	1	1		1		1	1	1
d02								0.6				
d03				1			0.5		1			
d04			0.9	0.3			0.6				0.2	
d05		1	0.9	0.5	1	1		1	1	0.9	1	
d06		1	0.8	0.5	1	1		1	1	1	0.9	
d07							0.3					
d08												
d09		0.8				0.3				0.6		
d10		1	0.6	0.5	1	0.8			1	0.6	0.5	
d11			0.1			0.3			1	0.6		
d12	0.9		0.5	0.2	1	0.5	0.4		0.2	0.5	0.3	
d13		0.8	0.8	0.5	1	1	0.8	1		0.6	0.5	
d14			0.2			0.5			1			
d15		0.3	0.1					0.8				
d16		0.9	0.5				0.5	0.6	1			
d17												
d18	0.6	1	0.8	1	1	1	0.9	0.8	1	1	0.9	1
d19		0.8	0.6	0.5	1	0.8	1	0.5	1	0.6	0.9	
d20												
d21		0.7	0.6		1	0.5				0.2		
d22							0.7	0.4	1			
d23												
d24				0.5							0.8	
d25												
d26												
d27												
d28		0.5	0.3		1	1		0.3	1	0.4		
d29												
d30												

- *time lag* (average interval between the search request and the answer)
- form of *presentation* of the output
- user *effort* in obtaining answers to the search requests
- *recall* of the system (proportion of relevant material actually retrieved in answer to a search request)
- *precision* of the system (proportion of retrieved material that is actually relevant)

The first four quantities are easily accessible. Recall and precision, however, are the real measures of the *effectiveness* of a retrieval system because they access the ability of the system to retrieve relevant documents while, at the same time, rejecting the nonrelevant ones. It is assumed

Table 4.7. User response for retrieval 5.
Query: Expert-System OR Logic-Programming

Document	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12
d01	0.5	0.3	0.2	1	1	0.2		0.8		0.8	0.5	1
d02												
d03												
d04												
d05		0.5					0.6			0.2		
d06		0.5				0.2	0.8			0.2		
d07												
d08												
d09		0.4					0.3			0.4		
d10		0.9				0.3	0.4		1	0.5		
d11						0.2	0.5			0.5		
d12						0.2				0.2	0.2	
d13		0.2					0.5					
d14												
d15		0.1										
d16		0.8	0.4				0.4		1			
d17												
d18		0.2	0.2	0.1		0.2	0.7			0.4		
d19		0.6			0.2	0.8			0.4			
d20												
d21		0.2										
d22												
d23												
d24												
d25												
d26												
d27												
d28		0.2				0.3						
d29												
d30												

that the more effective the system the more it will satisfy the user and that precision and recall are sufficient for the measurement of effectiveness Retrieval effectiveness, therefore, is measured by two parameters -- recall (R) and precision (P) defined as

$$R = \frac{\textit{number of relevant items retrieved}}{\textit{total number of relevant items in collection}}$$

$$P = \frac{\textit{number of relevant items retrieved}}{\textit{total number of items retrieved}}$$

4.3.1. Precision and Recall Based on the Retrieval Status Value

When the RSV is known, the measurement of the two parameters normally requires a threshold above which all documents having a higher RSV are considered relevant and are retrieved; those with a lower RSV are rejected. This is similar to the gist operation in GISTER in that it takes a gist level and produces a crisp answer. The relevance judged by a retrieval system then could be compared to the relevance substantiated by a user, based on the parameters defined above.

An alternative is to modify the definition of the parameters and to obtain the measure in two passes [Tong et al. 85]. In the first pass, the threshold is lowered until all the relevant (user judgement) documents are included in the retrieved set and the proportion of the relevant to retrieved documents is then counted. This is known as precision (with fixed recall equal to 1).

In the second pass, the threshold is raised until all nonrelevant documents are rejected and the retrieved relevant ratio is counted. This is recall (with precision equal to 1).

4.3.2. Performance Measure

The floating-threshold method was adopted here to determine the effectiveness of the system. Table 4.8 lists the measures for REDER. It is obtained based on the retrieval results displayed in Tables 4.2 through 4.7, and according to the method described in the previous section.

There are 12 users (u1 to u12) participating in the experiment. For each user, five retrieval activities were conducted, corresponding to the five queries (q1 to q5). For each retrieval, a pair of effectiveness measures were obtained; for example, pi and ri stand for the precision and recall, respectively, for the retrieval regarding the query qi.

The precision and recall were derived by comparing the result obtained from the user and that from REDER. Take the derivation of p1 and r1 for u6 as an example. These values were obtained based on the comparison of the column of data in Table 4.2, under q1, and that in Table 4.3, under u6. The floating-threshold method was applied to obtain the results: p1 = 0.93 and r1 = 0.96.

Table 4.8. Precision and recall for REDER.

Query	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10	u11	u12	Avg.
p1	0.73	0.87	0.96	0.7	1	0.93	0.83	0.87	0.87	0.83	0.83	0.93	0.876
r1	0.4	0.346	0.724	0.52	0.428	0.96	0.36	0.346	0.615	0.64	0.96	1	0.603
p2	0.43	0.4	0.67	0.37	0.3	0.43	0.43	0.37	0.37	0.4	0.4	0.2	0.398
r2	0.39	0.42	0.25	0.45	0.56	0.38	0.38	0.45	0.45	0.417	0.417	0.417	0.415
p3	0.167	1	0.2	1	1	0.23	0.17	1	1	1	0.17	0.067	0.583
r3	0.4	0.5	0.33	0.5	0.5	0.285	0.4	0.28	0.33	0.33	0.4	1	0.438
p4	0.1	0.4	0.53	0.367	0.33	0.43	0.3	0.37	0.4	0.4	0.33	0.06	0.334
r4	0.67	0.41	0.313	0.45	0.5	0.385	0.22	0.45	0.25	0.417	0.5	1	0.463
p5	1	0.4	0.13	0.067	1	0.26	0.3	1	0.067	0.3	0.067	1	0.466
r5	0.34	0.083	0.25	0.5	0.1	0.125	0.11	0.09	0.5	0.11	0.5	0.5	0.267

The results of this experiment demonstrate the feasibility of knowledge-based retrieval based on evidential reasoning. It can be concluded, therefore, that the system is able to produce satisfactory retrieval results for the users.

5. Conclusions and Recommendations

5.1. Conclusions

There are two facets in the problem of retrieval of information. The first is to obtain the relevant information from large amounts of documents, and the second is to determine where to start when large amount of relevant information has been retrieved. Research in this field has focused on the development of efficient methods for leading users to the relevant information.

The current trend is to apply AI techniques to solve the retrieval problem, especially knowledge-based systems for automatic query formulation and uncertainty reasoning for query evaluation. A Knowledge-based system has been proved to enhance the quality of the retrieval result; however, the adopted reasoning methods are either designed for special users only or they do not subsume the Boolean model.

As a result, the contribution of the research reported here was the development of a general framework for a theoretically sound knowledge-based retrieval system designed for general users, based on a process of evidential reasoning. Its induced *belief ranking principle* makes possible an interface to fuzzy models, thereby, bridging the gap between probabilistic and fuzzy models.

A prototype was implemented to demonstrate the capabilities of such a system based on evidential reasoning. The results of an actual retrieval experiment confirmed its enhanced performance.

5.2. Recommendations

This work provides a foundation for future extension and expansion of knowledge-based retrieval of information based on evidential reasoning. The introduction of fuzzy sets as the intermediate sets in the reasoning path could complicate the problem but would generalize the current approach.

Incorporating user feedback into the new system could produce some rules that will affect future retrieval results; user preference may also improve the efficiency of later retrievals. An alternative is *user modeling* before the actual retrieval and then tuning the results accordingly.

The retrieval of information among several information systems, each with its own knowledge base, could lead to automatic knowledge acquisition among cooperating expert systems. This would lead to system cooperation and knowledge sharing.

Appendix A: REDER Subsumes Boolean Model

This proof verifies that the knowledge-based retrieval system based on evidential reasoning as proposed in Section 3 subsumes the Boolean model of retrieval. The proof is divided into automatic query formulation, query evaluation, and retrieval-result assignment.

Automatic Query Formulation

In this proof, D denotes the set of documents, Q is any given Boolean query, T is the set of index (query) terms, and R is the retrieval result. It is assumed that Q has been polished by the thesaurus analysis.

In the new retrieval system, Q is expanded into a conceptual tree whose size is governed by the depth reasoning steps k which is a default value provided by the system or controlled by the user. On the other hand, because the Boolean model eliminates automatic query formulation, its reasoning process has only one step. To reduce the new system to the Boolean, k is set to 1 and the conceptual tree becomes Q.

Query Evaluation

As described in Section 1, any intermediate retrieval set of documents in the Boolean model is the result of set operations that reflect the Boolean connectives in the query. To reduce the new system, a pseudo-term p is introduced as the root of the conceptual tree and the belief measure is evenly distributed to each term in the query ti $(i = 1, \ldots, n)$. This measure is then acquired by the corresponding sets of documents as the initial value of their mass assignment

$$\begin{aligned} m(D_i) &= bpa(ti/p) \\ &= 1/n \end{aligned} \tag{A.1}$$

where D_i is the set of documents indexed by ti.

The next step is to prove that all the intermediate results (sets) of the two systems are the same except, in the new system, all of these results have retrieval status values (RSVs) in the range [0,1]. In the Boolean model, however, its RSVs are equal to 1.

The intermediate results are derived by applying the set connective to the indexed subsets of documents. If the connective between two terms ti and tj in the query is OR in the Boolean model, the resulting intermediate set Dij is the union of D_i and D_j

$$Dij = D_i \cup D_j \tag{A.2}$$

In the new system, the intermediate result is described by

$$RSV(dij) = \sum_{X \cap dij \neq \varnothing} m(X) \tag{A.3}$$

$$= m(D_i) \qquad \text{if } dij \in D_i \text{ and } dij \notin D_j$$

$$= m(D_i) + m(D_j) \qquad \text{if } dij \in D_i \cap D_j$$

$$= m(D_j) \qquad \text{if } dij \in D_j \text{ and } dij \notin D_i$$

$$= 0 \qquad \text{if } dij \notin D_i \cup D_j$$

If the connective is AND, the intermediate result in the Boolean model becomes

$$Dij = D_i \cap D_j \tag{A.4}$$

and, in the new system,

$$RSV(dij) = \frac{\sum_{D_i \cap D_j = X} m_1(D_i)\, m_2(D_j)}{1 - \sum_{D_i \cap D_j = \varnothing} m_1(D_i)\, m_2(D_j)} \tag{A.5}$$

$$= m(D_i)\, m(D_j) \qquad \text{if } dij \in D_i \cap D_j$$

$$= 0 \qquad \textit{otherwise}$$

Under both conditions, only those members in Dij have a nonzero RSV in the new system and the values obtained in Eqs. (A.3) and (A.5) are in the range $[0,1]$.

Retrieval-Result Assignment

In the final stage, the new system derives the result by taking those documents having a nonzero RSV as

$$R = \{dij \mid RSV(dij) \neq 0\}$$

and this result is the same as obtained by the Boolean model.

Q.E.D.

References

[Bartschi 85]

M. BARTSCHI, An Overview of Information Retrieval Subjects, *IEEE Computers 18(5)* (1985), 67-84.

[Biswas et al. 87]

G. BISWAS, J. C. BEZDEK, V. SUBRAMANIAN and M. MARQUES, Knowledge-Assisted Document Retrieval: II. The Retrieval Process, *Journal of the American Society for Information Science 38(2)* (1987), 97-110.

[Bonissone and Decker 85]

P. P. BONISSONE and K. S. DECKER, Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-off Precision and Complexity, *Proceedings of the Workshop on Uncertainty and Probability in Artificial Intelligence*, Los Angeles, 1985, 57-66.

[Bookstein 80]

A. BOOKSTEIN, Fuzzy Requests, *Journal of the American Society for Information Science 31(3)* (1980), 240-247.

[Bookstein 85]

A. BOOKSTEIN, Probability and Fuzzy Set Application to Information Retrieval, *Annual Review of Information Science and Technology 20* (1985), 117-151, Knowledge Industry Publications, Inc..

[Boose 85]

J. H. BOOSE, Personal Construct Theory and the Transfer of Human Expertise, *Advances in Artificial Intelligence*, 1985, 51-50.

[Buckland 83]

M. K. BUCKLAND, Relatedness, Relevance and Responsiveness in Retrieval Systems, *Information Processing & Management 19 (3)* (1983), 237-241.

[Buell and Kraft 81a]

D. A. BUELL and D. H. KRAFT, A Model for a Weighted Retrieval System, *Journal of the American Society for Information Science 32(3)* (1981), 211-216.

[Buell and Kraft 81b]

D. A. BUELL and D. H. KRAFT, Threshold Values and Boolean Retrieval Systems, *Information Processing and Management 17(3)* (1981), 127-136.

[Chen 88a]

L. A. CHEN, Information Retrieval as a Process of Evidential Reasoning, *Eighth International Conference in Computer Science*, Santiago, Chile, 1988, 99-108.

[Chen 88b]

L. A. CHEN, On Information Retrieval and Evidential Reasoning, *Technical Report UCB/Computer Science Dpt. 88/429* (1988), University of California, Berkeley.

[Chen 88c]

L. A. CHEN, Knowledge-Based Retrieval as the First Step of Knowledge Acquisition, *Third International Symposium on Knowledge Engineering*, Madrid, Spain, 1988, 173-182.

[Cleverdon, Mills and Keen 66]

C. W. CLEVERDON, J. MILLS and M. KEEN, Factors Determining the Performance of Indexing Systems, *ASLIB Cranfield Project*, Cranfield, 1966.

[Cooper 76]

W. S. COOPER, The Suboptimality of Retrieval Ranking Based on Probability of Usefulness, *Technical Report*, 1976.

[Cooper 88a]

W. S. COOPER, Getting Beyond Boole, *Information Processing & Management 24(3)* (1988), Pergamon Press.

[Cooper 88b]

W. S. COOPER, Notes on the Logical Design of Information Retrieval Systems, *Readings on the Principles of Information Retrieval*, 1988, 18-44.

[Dempster 67]

A. P. DEMPSTER, Upper and Lower Probabilities Induced by Multivalued Mapping, *Annals of Mathematical Statistics 38* (1967), 325-339.

[Feigenbaum 77]

E. A. FEIGENBAUM, The Art of Artificial Intelligence: I. Themes and Case Studies of Knowledge Engineering, Proceedings, *Fifth International Joint Conference on Artificial Intelligence*, 1977.

[Gordon and Shortliffe 84]

J. GORDON and E. H. SHORTLIFFE, Hypothesis Space, *Tech. Report HPP 84-35* (1984), Stanford University.

[Liu 87]

G. S. LIU, Causal and Plausible Reasoning in Expert Systems, *Ph.D. Dissertation*, Computer Science Division, University of California, Berkeley, 1987.

[Lowrance 82]

J. D. LOWRANCE, Dependency-Graph Models of Evidential Support, *Ph.D. Dissertation*, University of Massachusetts, Amherst, 1982.

[Lowrance and Garvey 82]

J. D. LOWRANCE and T. D. GARVEY, Evidential Reasoning: a Developing Concept, *Proceeding of the International Conference on Cybernetics and Society*, October 1982, 6-9.

[Lowrance 87]

J. D. LOWRANCE, Evidential Reasoning with Gister , *A Manual*, April 1987.

[Maron and Kuhns 60]

M. E. MARON and J. L. KUHNS, On Relevance, Probabilistic Indexing and Information Retrieval, *Journal of the Association for Computing Machinery 7(3)* (1960), 216-244.

[Radecki 76]

T. RADECKI, Mathematical Model of Information Retrieval Systems Based on the Concept of Fuzzy Thesaurus, *Information Processing and Management 12* (1976), 313-318.

[Radecki 79]

T. RADECKI, Fuzzy-Set Theoretical Approach to Document Retrieval, *Information Processing and Management 15(5)* (1979), 247-259.

[Rijsbergen 79]

C. J. V. RIJSBERGEN, *Information Retrieval*, Butterworths, London, 1979.

[Robertson and Sparck-Jones 76]

S. E. ROBERTSON and K. SPARCK-JONES, Relevance Weighting of Search Terms, *Journal of the American Society for Information Science 27(3)* (1976), 129-146.

[Robertson 77]

S. E. ROBERTSON, The Probability Ranking Principle in IR, *Journal of Documentation 33* (1977), 294-304.

[Robertson, Maron and Cooper 82]

S. E. ROBERTSON, M. E. MARON and W. S. COOPER, Probability of Relevance: A Unification of Two Competing Models for Document Retrieval, *Information Technology: Research and Development 1(1)* (1982), 1-21.

[Salton 71]

G. SALTON, *The SMART Retrieval System -- Experiments in Automatic Document Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[Salton and McGill 83]

G. SALTON and M. J. MCGILL, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.

[Salton 89]

G. SALTON, *Automatic Text Processing*, Addison-Wesley, 1989.

[Shafer 76]

G. SHAFER, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, 1976.

[Shafer 81]

G. SHAFER, Constructive Probability, *Synthesis 48* (1981), 1-60.

[Shafer 82]

G. SHAFER, Lindley's Paradox, *Journal of the American Statistical Association 77* (1982), 325-351.

[Shafer 87]

G. SHAFER, Implementing Dempster's Rule for Hierarchical Evidence, *Artificial Intelligence 33* (1987), 271-298.

[Tahani 76]

V. TAHANI, A Fuzzy Model of Document Retrieval Systems, *Information Processing and Management 12(3)* (1976), 177-188.

[Tong et al. 83]

R. M. TONG, D. G. SHAPIRO, J. S. DEAN and B. P. MCCUNE, A Comparison of Uncertainty Calculi in an Expert System for Information Retrieval, *International Joint Conference of Artificial Intelligence*, 1983, 194-197.

[Tong et al. 85]

R. M. TONG, V. N. ASKMAN, J. F. CUNNINGHAM and C. J. TOLLANDER, RUBRIC: An Environment for Full Text Information Retrieval, *on Research & Development in Information Retrieval*, 1985, 243-251.

[Tong and Shapiro 85]

R. M. TONG and D. G. SHAPIRO, Experimental Investigations of Uncertainty in a Rule-Based System for Information Retrieval, *International Journal of Man-Machine Studies 22* (1985), 265-282.

[Waterman and Hayes-Roth 82]

D. A. WATERMAN and F. HAYES-ROTH, *An Investigation of Tools for Building Expert Systems*, R2818-NSF, 1982.

[Wong and Yao 87]

S. K. M. WONG and Y. Y. YAO, A Statistical Similarity Measure, *Proceedings of the Tenth Annual International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1987, 3-12.

[Yen 86]

J. YEN, Evidential Reasoning in Expert Systems, *Ph.D. Dissertation*, Computer Science Division, University of California, Berkeley, 1986.

[Zadeh 65]

L. A. ZADEH, Fuzzy Sets, *Information and Control 8* (1965), 338-353.