FUZZY INFORMATION RETRIEVAL AND DATABASES

Information processing and management has become one of the topics that has stimulated great interest over the past several years. The technological advances in databases and retrieval systems and the ability to access such data over the Internet has focused developments in this area. Information systems are designed to model, store, and retrieve large amounts of information effectively. From a developmental point of view, the management of unstructured information (texts), on one hand, and structured information (formatted data representing factual business information), on the other, have given rise to two different lines of research and products: information retrieval systems and database management systems.

Being able to naturally handle the imprecision and vagueness that we experience in the real world of information systems is very desirable. Fuzzy set theory has proven to be a very powerful tool to handle this sort of uncertainty in many areas. In information systems, the two main issues in which uncertainty should be reflected are the representation scheme and the querying mechanism; these are discussed here.

FUZZY DATABASES

The earliest attempt to represent inexact data in databases was the introduction of the concept of null values by Codd (1). The first extensions of the relational data model that incorporated nonhomogeneous domain sets did not use fuzzy set theory. Rather, they attempted to represent null values and intervals. The ANSI/X3/SPARC report of 1975 (2), for instance, notes more than a dozen types of null. At one end of the spectrum, null means completely unknown. For example, a null value in the current salary of an employee could mean the actual value is any one of the permissible values for the salary domain set.

Without resorting to fuzzy measures, a user can specify some information about a value that further restricts it. A subset or range of values of the domain set may be described within which the actual attribute value must lie. The user or the system (via functional dependencies) may specify subsets or subranges within which the actual value must not lie. Yet another option is to label null values in a manner that requires distinct nulls in different portions of the database to have a particular actual value relationship (usually equality) if they have the same label. The semantics of the null value range from "unknown" (e.g., the current salary of an employee) to "not applicable" (e.g., subassembly number of a part that is not a subassembly) to "does not exist" (e.g., middle name of a person). These last two meanings, however, are not related to uncertainty.

Codd proposes a three-value logic using T, F, and $\perp$ (null in the sense of unknown) in conjunction with the following predicates:

$$X\,\theta\,Y \equiv \perp \text{ if } X \text{ or } Y \text{ is null and } \theta \text{ is } <, \leq, =, \neq, \geq, >$$
$$\perp \in S \equiv \perp \text{ for any set } S$$
$$S \supseteq \{\perp\} \equiv \perp \text{ for any set } S$$

There is a problem. Because of the variety of meanings possible for null values, they cannot discriminate well enough (i.e., they are "overloaded" in the programming language sense). Two possible solutions are to maintain multiple nulls or to provide semantic interpretation external to the database.

Range Values Approach

As discussed, it is possible to have a variety of nulls with different semantics. However, these are not adequate to represent the possibility of a range of values. For example, we may not know exactly the age of a house, but we know it is in the range of 20 to 30 years. So we have an interval of values and know one is correct but do not necessarily know exactly which one.

An early development in this area by Grant (3) extended the relational model to allow range values. Basically three types of values are allowed: a single number for the case of complete information; a pair of numbers (i, j) $[i \leq j]$ for the case of partial information in the form of a range of possible values; and finally a null value in the case of no information. To deal with comparisons of such values for purposes of defining relations and relational operators, true and maybe predicates are defined where the maybe predicate means that it is true or maybe true. For example, consider a relation R with three tuples. For an attribute Years, the values for each tuple are: 15; 8; (20,30). It is definite that $15 \in R$, but it is not certain if 25 is in R, so we have $25 \in_{\mathrm{M}}$ (maybe an element of) R. Note that, by the definition of the maybe predicate, we also have $15 \in_{\mathrm{M}} R$.

The basis of the relational model is set theoretic, so we can view a relation as a set of tuples. In a set there should not normally be duplicate elements, and the issue of elimination of duplicate tuples plays a significant role in inexact and imprecise models of data. For several fuzzy database models, the elimination of redundant tuples requires careful consideration.

In the case of a range of values, we can see some of the issues that will arise in the case of fuzzy databases. In particular, duplicate tuples are allowed because, even if they appear to be identical, they may actually stand for different values (i.e., have different interpretations). Consider the possibility of the tuple (20,30) appearing twice in the preceding relation R. In one case, it may stand for the actual value 25 and in the other, 28. If the set of possible interpretations for this range comprises the 11 values: 20, 21, . . ., 30, then there can be at most 11 occurrences of the tuple value (20,30) without violating the "no duplicate tuples" rule.

Lipski's Generalized Approach to Uncertainty

Lipski (4) proposed a more general approach. He does not, for instance, assume that null means that a value is completely unknown. Given that there may be labeled or restricted value nulls, let $\|Q\|$ denote all real-world objects that a query Q could represent. Let T be a database object and $\|T\|$ be all real-world objects it could represent. These are also known as external and internal interpretations.

Assume a relation EMPLOYEE with domains NAME and AGE. The database object T = [Bob 30–35] could represent six real-world objects (one for each year in the age range). A query Q places each database object in one of three categories.

$$\begin{array}{ll} T \in \{\text{surely set}\} & \text{if } \|Q\| \supset \|T\| \\ T \in \{\text{possible set}\} & \text{if } \|Q\| \cap \|T\| \neq \Phi \\ T \in \{\text{eliminated set}\} & \text{if } \|Q\| \cap \|T\| = \Phi \end{array}$$

For instance, the query, EMPLOYEE [AGE $>$ 32], places T in the possible set, while EMPLOYEE [AGE $>$ 25] $\vee$ EMPLOYEE [AGE $<$ 40] places T in the surely set. The first two categories are also known as the lower value $\|Q\|_*$ and upper value $\|Q\|^*$, and these limiting interpretations are characterized in this approach. A number of relationships that assist in evaluating this sort of query have been developed. It should be noted that because the representation of inexact data is sufficiently generalized, it becomes intimately related to the uncertainty data modeling using fuzzy sets, which we will be describing shortly.

Statistical and Probabilistic Databases

The main work in the area of statistical approaches is that of Wong (5) in which he handles a large class of uncertainty cases by statistical inference. This formulation approaches the uncertainty of the real-world data by assuming an ideal world of perfect information to which the incomplete data may be statistically compared. The prior information from this comparison is represented either as a distortion function or a conditional distribution. Missing and combined attributes can be dealt with by distortion functions.

The more direct method of dealing with uncertainty and incompleteness is to specifically use a probabilistic data model, and the most completely developed approach is that in which probabilities are associated with the values of the attributes (6). In this model, because each stochastic attribute is treated as a discrete probability distribution function, the probabilities for each attribute (in a tuple) must be normalized (sum to 1.0). However, it may be difficult to ascertain exact probabilities for all possible domain values. As a result, they developed the concept of missing probabilities to account for such incompletely specified probability distributions. It permits the probabilistic model to capture uncertainty in data values as well as in the probabilities. When updating or entering data into a probabilistic relation, it is not necessary to have all information before some tuple can be entered, allowing a natural use of such uncertain information.

Fuzzy Databases Models of Imprecision

The relational model has been the dominant database model for a considerable period of time, and so it was naturally used by researchers to introduce fuzzy set theory into databases. Much of the work in the area has been in extending the basic model and query languages to permit the representation and retrieval of imprecise data. A number of related issues such as functional dependencies, security, and implementation considerations have also been investigated (7).

Two major approaches have been proposed for the introduction of fuzziness in the relational model. The first one uses the principle of replacing the ordinary equivalence among domain values by measures of nearness such as similarity relationships (8), proximity relationships (9), and distinguishability functions (10). The second major effort involves a variety of approaches that directly use possibility distributions for attribute values (11,12). There have also been some mixed models combining these approaches (13,14).

We can also characterize these approaches relative to their extensions of the relational model. As we have seen in capturing incompleteness or uncertainty, it is necessary to extend the basic relational model by using non-first normal forms. In the first approach using nearness measures, the imprecision of the actual data values is implicit, using a separate relation or table for the similarity or proximity relationship. Generally with the use of possibility distributions, most approaches have some imprecise description of the data explicitly or directly represented in the basic attribute values of the relation. We characterize these approaches as being either homogeneous or heterogeneous representations.

The distinguishing characteristic of an ordinary relational database (or ordinary databases of other forms) is the uniformity or homogeneity of the represented data (15). For each domain, there is a prescribed set of values from which domain values may be selected. Furthermore, each element of the domain set is of the same structure (e.g., integers, real numbers, or character strings). With the use of similarity or proximity relationships, the imprecision in domain values is implicit, and so the representation remains homogeneous. These approaches are thus closer to ordinary crisp relational models and can be shown to have properties that closely follow those of conventional relational models.

To more directly represent uncertainty within the domain values themselves requires departure from homogeneity of representation. These models based on possibility theory provide the ability to model more forms of uncertainty. As would be expected from the increased power of representation, there is a tradeoff in more complexity of implementation. The more complex extensions of the basic relational model lead us to classify them using a heterogeneous representation. This is just a matter of degree, and some approaches may be more heterogeneous than others.

Membership Values Models. The simplest form for a fuzzy database is the attachment of a membership value (numeric or linguistic) to each tuple. This permits maintenance of homogeneous data domains and strongly typed data sets. However, the semantic content of the fuzzy membership domain is used during query processing. We will consider examples that illustrate two distinct semantics for the membership domain. In the first relation, Investment_Sites, we have tuples with attributes of [site-id, classification, membership value]: {[12, residential-1, 1.0], [14, residential-2, 0.7], [79, light-commercial, 0.85], . . .}. The membership value here denotes the degree to which the tuple belongs within the relation (16).

The second example is the relation Resume_Analysis, which represents the analysis criteria of potential employees: {[physics, science, 1.0], [botany, science, 0.7], [statistics, analysis, 0.8], . . .}. In the relation, the membership value denotes the strength of the dependency between the key attribute, Subject, and the attribute Classification (17).

Similarity-Based Fuzzy Models. In the late 1970s, Buckles and Petry (8) were the first to use similarity relationships in a relational model. Their approach attempted to generalize the concept of null and multiple-valued domains for implementation within an operational environment consistent with the relational algebra. In fact, the nonfuzzy relational database is a special case of their fuzzy relational database approach.

For each domain j in a relational database, a domain base set D_j is understood. Domains for fuzzy relational databases are either discrete scalars or discrete numbers drawn from either a finite or infinite set. An example of a finite scalar domain is a set of linguistic terms. For example, consider a set of terms that can be used for subjective evaluation of a patient's health: {critical, severe, poor, so-so, average, good, excellent}. The fuzzy model uses a similarity relationship to allow the comparison of these linguistic terms. The domain values of a particular tuple may also be single scalars or numbers (including null) or a sequence of scalars or numbers. Consider, for example, the assessments made in the triage database to permit ranking of patient treatment. If we include linguistic descriptions of the severity of patients and combine these with procedure time estimates, we have tuples in the relation such as: {[p1, {so-so, average}, {20, 30}], [p2, poor, {20, 50}], [p3, {poor, severe}, {80–120}], . . .}

The identity relation used in nonfuzzy relational databases induces equivalence classes (most frequently singleton sets) over a domain D, which affects the results of certain operations and the removal of redundant tuples. The identity relation is replaced in this fuzzy relational database by an explicitly declared similarity relation (18) of which the identity relation is a special case. A similarity relation $s(x, y)$ for given domain D is a mapping of every pair of elements in the domain onto the unit interval [0, 1] with the following three properties, $x, y, z \in D$:

1. Reflexive: $s_D(x, x) = 1$
2. Symmetric: $s_D(x, y) = s_D(y, x)$
3. Transitive: $s_D(x, z) \geq \mathrm{Max}(\mathrm{Min}[s_D(x, y), s_D(y, z)])$

Next the basic concepts of fuzzy tuples and interpretations must be described. A key aspect of most fuzzy relational databases is that domain values need not be atomic. A domain value d_i, where i is the index of the attribute in the tuple, is defined to be a subset of its domain base set D_i. That is, any member of the power set may be a domain value except the null set. Let $\mathbf{P}(D_i)$ denote the power set of $D_i - \emptyset$.

A *fuzzy relation* R is a subset of the set cross product $\mathbf{P}(D_1) \times \mathbf{P}(D_2) \times \cdots \times \mathbf{P}(D_m)$. Membership in a specific relation r is determined by the underlying semantics of the relation. For instance, if D_1 is the set of major cities and D_2 is the set of countries, then (Paris, Belgium) $\in \mathbf{P}(D_1) \times \mathbf{P}(D_2)$—but it is not a member of the relation A (capital-city, country).

A *fuzzy tuple* t is any member of both r and $\mathbf{P}(D_1) \times \mathbf{P}(D_2) \times \cdots \times \mathbf{P}(D_m)$. An arbitrary tuple is of the form $t_i = [d_{i1}, d_{i2}, \ldots, d_{im}]$ where $d_{ij} \subseteq D_j$.

An *interpretation* $\alpha = [a_1, a_2, \ldots, a_m]$ of a tuple $t_i = [d_{i1}, d_{i2}, \ldots, d_{im}]$ is any value assignment such that $a_j \in d_{ij}$ for all j.

In summary, the space of interpretations is the set cross product $D_1 \times D_2 \times \cdots \times D_m$. However, for any particular relation, the space is limited by the set of valid tuples. Valid tuples are determined by an underlying semantics of the relation. Note that in an ordinary relational database, a tuple is equivalent to its interpretation.

Some aspects of the max-min transitivity in a similarity can cause difficulty in modeling the relationship between domain elements. It can be difficult to formulate the transitive property of the relationship correctly. Furthermore at some α level, domain elements only weakly related can be forced together in a merged set of retrieved values. The essential characteristic that produces the desirable properties of uniqueness and well-defined operations is *partitioning* of the attribute domains by the similarity relationship.

Shenoi and Melton (9) show how to use proximity relations (nontransitive) for the generation of partitions of domains. The fuzzy relational model is extended by replacing similarity relations with proximity relations on the scalar domains. Recall that a proximity relation $P(x, y)$ is reflexive and symmetric but not necessarily transitive. This can also be related to a more generalized approach to equivalence relations for a fuzzy database model (19).

Possibility Theory-Based Database Models. In the possibility theory-based approach (11,20), the available information about the value of a single-valued attribute A for a tuple t is represented by a possibility distribution $\pi_{A(t)}$ on $D \cup \{e\}$ where D is the domain of the attribute A and e is an extra-element that stands for the case when the attribute does not apply to t. The possibility distribution $\pi_{A(t)}$ can be viewed as a fuzzy restriction of the possible value of $A(t)$ and defines a mapping from $D \cup \{e\}$ to [0, 1]. For example, the information "Paul has considerable experience" ($\pi_{e(p)}$) will be represented by ($\forall d \in D$):

$$\pi_{e(p)}(e) = 0 \quad \text{and} \quad \pi_{e(p)}(d) = \mu_{\mathrm{c}}(d)$$

Here μ_c is a membership function that represents the vague predicate "considerable" in a given context, such as the number of years of experience or the number of years of education.

It is important to notice that the values restricted by a possibility distribution are considered as mutually exclusive. The degree $\pi_{A(t)}(d)$ rates the possibility that $d \in D$ is the correct value of the attribute A for the tuple t. Note that $\pi_{A(t)}(d) = 1$ only means that d is a completely *possible* value for $A(t)$, but it does not mean that it is certain that d is the value of A for the tuple (or in other words that d is *necessarily* the value of A for t), unless

$$\forall d' \neq d, \pi_{A(t)}(d') = 0$$

Moreover, the possibility distribution $\pi_{A(t)}$ should be normalized on $D \cup \{e\}$ (i.e., $\exists d \in D$ such that $\pi_{A(t)}(d) = 1$ or $\pi_{A(t)}(e) = 1$). This means that it must be the case that at least one value of the attribute domain is completely possible or that the attribute does not apply. The following null value situations may be handled in this framework:

1. Value of A for t is completely unknown: $\forall d \in D$, $\pi_{A(t)}(d) = 1$, $\pi_{A(t)}(e) = 0$.
2. The attribute A does not apply for the tuple t: $\forall d \in D$, $\pi_{A(t)}(d) = 0$, $\pi_{A(t)}(e) = 1$.

3. It is not clear whether situation 1 or 2 applies: $\forall d \in D$, $\pi_{A(t)}(d) = 1$, and $\pi_{A(t)}(e) = 1$.

Thus, such an approach is able to represent, in a unified manner, precise values (represented by singletons), null values, and ill-known values (imprecise ones represented by crisp sets or vague ones represented by fuzzy sets). In this approach, multiple-valued attributes can be formally dealt with in the same manner as single-valued ones, provided that possibility distributions defined on the power set of the attribute domains rather than on the attribute domains themselves are used. Indeed, in the case of multiple-valued attributes, the mutually exclusive possibilities are represented by subsets of values.

Possibility and Necessity Measures. If two values a and b are described by their respective possibility distributions π_a and π_b, then they can be compared according to the extension principle (21). This leads to two degrees, expressing the extent to which the values *possibly* and *necessarily* satisfy the comparison relation. For equality, these degrees are given by

$$\begin{aligned}\mathrm{poss}(a = b) &= \sup_{x,y}(\min(\pi_a(x), \pi_b(y), \mu = (x, y)))\\ \mathrm{nec}(a = b) &= 1 - \sup_{x,y}(\min(\pi_a(x), \pi_b(y), \mu \neq (x, y)))\\ &= \inf_{x,y}(\max(1 - \pi_a(x), 1 - \pi_b(y), \mu = (x, y)))\end{aligned}$$

Of course, when a and b are precisely known, these two degrees collapse (and take their value in {0, 1}) because there is no uncertainty. Otherwise, the fact that two attribute values (in the same tuple or in two distinct tuples) are represented by the same possibility distribution does not imply that these values must be equal. For instance, if John's experience is "considerable" and Paul's experience is also "considerable," John and Paul may still have different amounts (e.g., years) of experience. This point is just a generalization of what happens with null values (if John's experience and Paul's experience are completely unknown, both are represented by a null value, whatever its internal representation, even though their years of experience are potentially distinct). The equality of two incompletely known values must be made explicit and could be handled in the relational model in extending the notion of marked nulls.

Querying Fuzzy Relational Databases

In systems that are relationally structured and use fuzzy set concepts, nearly all developments have considered various extensions of the relational algebra. Its syntactic structure is modified to the extent that additional specifications are required. Use of the relational calculus with a similarity model has also been studied (22). The relational calculus provides a nonprocedural specification for a query and can be extended more easily to a higher-level query language.

Similarity-Based Querying. To illustrate the process of query evaluation for similarity databases, we examine a generalized form of Boolean queries that may also be used to retrieve information (23). The details of query evaluation can be seen more easily in this sort of query.

A query Q (a_i, a_h, . . ., a_k) is an expression of one or more factors combined by disjunctive or conjunctive Boolean operators: V_i op V_h op $\cdots$ op V_k. In order to be well formed with respect to a relation r having domain sets D_1, D_2, . . ., D_m, each factor V_j must be

1. a domain element a, $a \in D_j$, where D_j is a domain set for r, or
2. a domain element modified by one or more linguistic modifiers (e.g., NOT, VERY, MORE-OR-LESS).

The relation r may be one of the original database relations or one obtained as a result of a series of fuzzy relational algebra operations. Fuzzy semantics apply to both operators and modifiers. An example query is

MORE-OR-LESS big *and* NOT VERY VERY heavy

where "big" is an abbreviation of the term (SIZE = big) in a relation having domain called SIZE. The value "heavy" is likewise an abbreviation. The linguistic hedge VERY can be interpreted as CON(F), concentration, and MORE-OR-LESS as DIL(F), dilation.

A membership value of a tuple in a response relation r is assigned according to the possibility of its matching the query specifications. Let $a \in D_j$ be an arbitrary element. The membership value $\mu_a(b)$, $b \in D_j$, is defined based on the similarity relation $s_j(a, b)$ over the domain. The query $Q(\cdot)$ induces a membership value $\mu_Q(t)$ for a tuple t in the response r as follows:

1. Each interpretation $I = [a'_1, a'_2, \ldots, a'_m]$ of t determines a value $\mu_{a_j}(a'_j)$ for each domain element a_j, of Q (a_i, a_h, . . ., a_k).
2. Evaluation of the modifiers and operators in $Q(\cdot)$ over the membership values $\mu_{a_j}(a'_j)$ yields $\mu_Q(I)$, the membership value of the interpretation with respect to the query.
3. Finally, $\mu_Q(t) = \max_{I \text{ of } t}\{\mu_Q(I)\}$.

In short, the membership value of a tuple represents the best matching interpretation. The response relation is then the set of tuples having nonzero membership values. In practice, it may be more realistic to consider only the tuple with the highest value.

Possibility-Based Framework for Querying. There are several approaches for querying relational databases where some incompletely known attribute values are represented by possibility distributions. One may distinguish between an approach that is set in a pure possibilistic framework (11) (approximate reasoning under uncertainty) and others that do not use such a strict theoretic framework (24–26).

According to the possibilistic view (11), when a condition applies to imperfectly known data, the result of a query evaluation can no longer be a single value. Because the precise values of some attributes for some items are not known, the fact that these items do or do not satisfy the query (to some degree) may be uncertain. This is why the two degrees attached to two points of view are used: the extent to which it is **possible** (resp. **certain**) that the condition is satisfied. From the possibility distributions $\pi_{A(t)}$ and a subset P (ordinary or fuzzy), one can compute the fuzzy set ΠP (resp. NP) of the items whose A-value possibly (resp. necessarily) satisfies the condition P. The membership degrees of a tuple t to ΠP and NP are, respectively, given by (27)

$$\begin{aligned}
\mu_{\Pi P}(t) &= \Pi(P; A(t)) = \sup_{d\in D}\min(\mu_P(d), \pi_{A(t)}(d)) \\
\mu_{NP}(t) &= \mathrm{N}(P; A(t)) = 1 - \Pi(\overline{P}; A(t)) \\
&= 1 - \sup_{d\in D\cup\{e\}}\min(\mu_{\overline{P}}(d), \pi_{A(t)}(d)) \\
&= \inf_{d\in D\cup\{e\}}\max(\mu_P(d), 1 - \pi_{A(t)}(d))
\end{aligned}$$

$\Pi(P;\ A(t))$ estimates to what extent at least one value restricted by $\pi_{A(t)}$ is compatible with P, and $N(P;\ A(t))$ estimates to what extent all the values more or less possible for $A(t)$ are included in P. It can be shown that ΠP and NP always satisfy the inclusion relation $\Pi P \supseteq NP$ (i.e., $\forall t,\ \mu_{NP}(t) \leq \mu_{\Pi P}(t)$), provided that $\pi_{A(t)}$ is normalized.

If John's age and the fuzzy predicate "middle-aged" are represented according to a possibility distribution, the evaluation of the condition: John's age = "middle-aged" is based on the computation of the values:

$$\min(\pi_{ja}(u), \mu_{ma}(u)) \quad \text{and} \quad \max(1 - \pi_{ja}(u), \mu_{ma}(u))$$

Thus, in case of incomplete information, it is possible to compute the set of items that more or less possibly satisfy an elementary condition and to distinguish the items that more or less certainly satisfy this condition.

FUZZY INFORMATION RETRIEVAL

Information retrieval systems (IRS) are concerned with the representation, storage, and accessing of a set of documents. These documents are often in the form of textual information items or records of variable length and format, such as books and journal articles (28). The specific aim of an IRS is to evaluate users' queries for information based on a content analysis of the documents stored in the archive. In response to a user query, the IRS must identify what documents deal with the information being requested via the query and retrieve those that satisfy the query.

Fuzzy IR models have been defined to overcome the limitations of the crisp Boolean IR model so as to deal with

1. discriminated (and possibly ranked) answers reflecting the variable relevance of the documents with respect to queries
2. imprecision and incompleteness in characterizing the information content of documents
3. vagueness and incompleteness in the formulation of queries

Fuzzy extended Boolean models constitute a superstructure of the Boolean model by means of which existing Boolean IRSs can be extended without redesigning them completely. The softening of the retrieval activity in order to rank the retrieved items in decreasing order of their presumed relevance to a user query can greatly improve the effectiveness of such systems. This objective has been approached by extending the Boolean models at various levels.

1. Fuzzy extension of document representation—The aim here is to provide more specific and exhaustive representations of the documents' information content in order to lower the imprecision and incompleteness of the Boolean indexing. This is done by incorporating significance degrees, or index term weights, in the representation of documents (29).
2. Fuzzy generalization of Boolean query language—The objective here is to render the query language more expressive and natural than crisp Boolean expressions in order to capture the vagueness of user needs as well as simplify the user system interaction. This is carried out at two levels. The first is through the definition of more expressive, as well as soft, selection criteria that allow the specification of different importance levels of the search terms. Query languages based on numeric query term weights with different semantics have been presented as an aid to define more expressive selection criteria (30,31). Also, an evolution of these approaches introduced linguistic query weights specified by fuzzy variables (e.g., important or very important) to express different levels of importance for query terms (32).

Incorporating fuzzy representations for documents in a Boolean IRS is a sufficient condition to improve the system with the ranking ability. As a consequence of this extension, the exact matching applied by a Boolean system can be softened to a partial matching mechanism, evaluating, for each document, the anticipated degree of satisfaction of the document with regard to a user's query. The value thus generated is called a retrieval status value (RSV) and is used as the basis for ranking the documents. This ranking is used for retrieval and display of those documents.

Fuzzy knowledge-based IRS models have been defined to index and retrieve documents in specific subject areas. To date, it has been found that IRSs are not adequate to deal with general collections. Reference 33 uses rules to represent semantic links between concepts; the nature of the links (e.g., synonymous terms, broader terms, narrower terms) and the strength of the links (represented by weights) are stored in the knowledge base and are defined by experts in the field. This is used to expand the query evaluation, by applying an inference process that allows one to find information that the user did not explicitly request but that is deemed "likely" to be of interest.

Fuzzy Indexing Procedures

In an information retrieval system, the generation of a representation of each document's subject content is called indexing. The basic problem is to capture and synthesize the meaning of a document written in natural language. In defining an indexing procedure (which can be either manual or automatic), one must first consider retrieval performance, via a document representation that allows the IRS to be able to retrieve all the relevant documents and none of the nonrelevant documents in response to a user query and then also consider exhaustivity (describing fully all aspects of a document's contents).

The Boolean retrieval model can be associated with automatic text indexing. This model provides a crisp representation of the information content of a document. A document is formally represented by the set of its index terms:

$$R(d) = \{t | t \in T, F(d, t) > 0\} \quad \text{for } d \in D$$

in which the indexing (membership) function F correlating terms and documents is restricted to {0, 1}. Of course, $F(d, t) = 1$ implies the presence of term t in document d; and $F(d, t) = 0$ implies the absence of the term in the document.

To improve the Boolean retrieval with a ranking ability, the Boolean representation has been extended within fuzzy set theory by allowing the indexing function F to take on values in the unit interval [0, 1]. Here, the index term weight $F(d, t)$ represents the degree of significance of the concept as represented by term t in document d. This value can be specified between no significance [$F(d, t) = 0$] and full significance [$F(d, t) = 1$] and allows a ranking of the retrieval output, providing improved user satisfaction and system performance. Consequently, a document is represented as a fuzzy set of terms

$$R(d) = \{\langle t, \mu_{d(t)} \rangle | t \in T\} \quad \text{for } d \in D$$

in which $\mu_{d(t)} = F(d, t)$. This implies that F is a fuzzy set membership function, measuring the degree to which term t belongs to document d (34). Through this extension, the retrieval mechanism can compute the estimated relevance of each document relative to the query, expressed by a numeric score called a retrieval status value. The RSV denotes how well a document seemingly satisfies the query (35,36). The definition of the criteria for an automatic computation of $F(d, t)$ is a crucial aspect; generally this value is defined on the basis of statistical measurements with the aim of optimizing retrieval performance.

Fuzzy Associations. Another concept linked to automatic indexing to enhance the retrieval of documents is that based on fuzzy associations, named fuzzy associative information retrieval models (37–40). These associative information retrieval models work by retrieving additional documents that are not indexed by the terms in a given query but are indexed by other terms, associated descriptors, that are related to the query terms.

Fuzzy association in information retrieval generally refers to the use of fuzzy thesauri where the links between terms are weighted to indicate strength of association. Moreover, this notion includes generalizations such as fuzzy pseudothesauri (41) and fuzzy associations based on a citation index (42). Ogawa et al. (43) propose a keyword connection matrix to represent similarities between keywords so as to reduce the difference between relationship values initially assigned using statistical information and a user's evaluation.

Generally, a fuzzy association between two sets $X = \{x1, \ldots, xm\}$ and $Y = \{y1, \ldots, yn\}$ is formally defined as a fuzzy relation:

$$f : X \times Y \to [0, 1]$$

By varying the semantics of the sets X and Y in information retrieval, different kinds of fuzzy associations can be derived.

Fuzzy Querying

Two factors have been independently taken into account to extend the Boolean query language, making the selection criteria more powerful, and softening and enriching the aggregation operators. First, consider the basic query processing model.

The main aim in extending the selection criteria is to provide users with the possibility of specifying differing importances of terms in order to determine which documents should be relevant. This has been achieved by preserving the Boolean structure of the query language and by associating with each term a numeric value to synthesize importance.

Now, let's define for Q = {a set of user queries for document}, $\mathbf{a}(q, t)$: $Q \times T \to [0, 1]$, where $\mathbf{a}(q, t)$ is the importance of term t in describing the query q and is called a query term weight. It is here that one begins to introduce problems in terms of maintaining the Boolean lattice (44). Because of that, certain mathematical properties can be imposed on F, but more directly on $\mathbf{a}$ and on the matching procedure. Moreover, there is a problem in developing a mathematical model that will preserve the semantics (i.e., the meaning) of the user query. The weight $\mathbf{a}$ can be interpreted as an importance weight, as a threshold, or as a description of the "perfect" document.

Let g: $[0, 1] \times [0, 1] \to [0, 1]$ [i.e., $g(F, \mathbf{a})$ is the RSV for a query q of one term t, with query weight $\mathbf{a}$, with respect to a given document d, which has index term weight $F(d, t)$ for the same term t]. This function g can be interpreted as the evaluation of the document in question along the dimension of the term t if the actual query has more than one term.

It has been suggested that terms be evaluated from the bottom up, evaluating a given document against each term in the query and then combining those evaluations according to the query structure (45). Reference 46 shows that this criterion for a g function, called separability (47), preserves a homomorphism between the document evaluations for single-term queries and the document evaluations for complex Boolean queries.

A first formulation of the g function treats the a values as relative importance weights; for example, one could specify $g = F^*a$. However, this can lead to problems, such as when using an AND (44). In this case, a very small value of a for one of the terms in an AND query will dominate the min function and force a decision based on the least important (smallest a) term, which is just the opposite of what is desired by the user. This problem is precisely what prompted some researchers to consider g functions that violate separability (31,48).

To achieve consistency in the formalization of weighted Boolean queries, some approaches do not maintain all the properties of the Boolean lattice: Kantor (49) generates a mathematical formulation of the logical relationships between weighted queries, using a vapid query with all zero weights.

FUTURE DIRECTIONS

Several specialized aspects not covered in this article are of increasing research importance. Fuzzy functional dependencies relate to several issues for fuzzy databases including database design and integrity management (50,51). The actual application of uncertainty in deployed database systems is following two directions. The first is the addition of uncertainty in object oriented databases (52,53). This is due to newer developments in object-oriented databases and their inherent capabilities such as encapsulated methods. Another direction is that of fuzzy-front end querying (54,55). This approach allows a general use with existing databases and also permits fuzzy querying of crisp data. A good general survey of some of the issues in these directions is (56).

BIBLIOGRAPHY

1. E. Codd, Extending the database relational model to capture more meaning, *ACM Trans. Database Sys.,* **4** (2): 156–174, 1979.
2. Anonymous, American National Standards Institute study group on database management: Interim report, *ACM SIGMOD Rec.,* **7**: 25–56, 1975.
3. J. Grant, Incomplete information in a relational database, *Fundamenta Informaticae,* **3** (4): 363–378, 1980.
4. W. Lipski, On semantic issues connected with incomplete information databases, *ACM Trans. Database Syst.,* **4** (3): 262–296, 1979.
5. E. Wong, A statistical approach to incomplete information in database systems, *ACM Trans. Database Systems,* **7** (4): 479–488, 1982.
6. D. Barbara, H. Garcia-Molina, and D. Porter, The management of probabilistic data, *IEEE Trans. Knowl. Data Eng.,* **4** (4): 487–502, 1992.
7. F. Petry, *Fuzzy Databases: Principles and Applications,* Boston: Kluwer, 1996.
8. B. Buckles and F. Petry, A fuzzy model for relational databases, *Fuzzy Sets and Systems,* **7** (3): 213–226, 1982.
9. S. Shenoi and A. Melton, Proximity relations in fuzzy relational databases, *Fuzzy Sets and Systems,* **31** (2): 287–296, 1989.
10. M. Anvari and G. Rose, Fuzzy relational databases, in J. Bezdek (ed.), *The Analysis of Fuzzy Information Vol. II,* Boca Raton FL: CRC Press, 1987, pp. 203–212.
11. H. Prade and C. Testemale, Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries, *Information Sci.,* **34** (2): 115–143, 1984.
12. M. Zemankova and A. Kandel, Implementing imprecision in information systems, *Information Sci.,* **37** (1): 107–141, 1985.
13. E. Rundensteiner, L. Hawkes, and W. Bandler, On nearness measures in fuzzy relational data models, *Int. J. Approximate Reasoning* **3** (4): 267–298, 1989.
14. J. Medina, O. Pons, and M. Vila, Gefred: A generalized model to implement fuzzy relational databases, *Information Sci.,* **47** (5): 234–254, 1994.
15. B. Buckles and F. Petry, Uncertainty models in information and database systems, *J. Information Sci.: Principles and Practice,* **11** (1): 77–87, 1985.
16. C. Giardina, Fuzzy databases and fuzzy relational associative processors, *Technical Report,* Hoboken NJ: Stevens Institute of Technology, 1979.
17. J. Baldwin, Knowledge engineering using a fuzzy relational inference language, *Proc IFAC Symp. on Fuzzy Information Knowledge Representation and Decision Analysis,* pp. 15–21, 1983.
18. L. Zadeh, Similarity relations and fuzzy orderings, *Information Sci.,* **3** (3): 177–200, 1971.
19. S. Shenoi and A. Melton, An extended version of the fuzzy relational database model, *Information Sci.,* **51** (1): 35–52, 1990.
20. H. Prade, Lipski's approach to incomplete information databases restated and generalized in the setting of Zadeh's possibility theory, *Information Systems,* **9** (1): 27–42, 1984.
21. L. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems,* **1** (1): 3–28, 1978.
22. B. Buckles, F. Petry, and H. Sachar, A domain calculus for fuzzy relational databases, *Fuzzy Sets and Systems,* **29** (4): 327–340, 1989.
23. B. Buckles and F. Petry, Query languages for fuzzy databases, in J. Kacprzyk and R. Yager (eds.), *Management Decision Support Systems Using Fuzzy Sets and Possibility Theory,* Koln GR: Verlag TUV Rheinland, 1985, pp. 241–252.
24. M. Umano, Retrieval from fuzzy database by fuzzy relational algebra, in E. Sanchez and M. Gupta (eds.), *Fuzzy Information, Knowledge Representation and Decision Analysis,* New York: Pergamon, 1983, pp. 1–6.
25. Y. Takahashi, A fuzzy query language for relational database, *IEEE Trans. Syst. Man. Cybern.,* **21** (6): 1576–1579, 1991.
26. H. Nakajima, T. Sogoh, and M. Arao, Fuzzy database language and library—fuzzy extension to SQL, *Proc. Second International Conference on Fuzzy Systems,* Los Alamitos, CA: IEEE Computer Society Press, 1993, pp. 477–482.
27. D. Dubois and H. Prade (with the collaboration of H. Farreny, R. Martin-Clouaire, and C. Testemale), *Possibility Theory: An Approach to Computerized Processing of Uncertainty,* New York: Plenum, 1988.
28. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval,* New York: McGraw-Hill, 1983.
29. T. Radecki, Fuzzy set theoretical approach to document retrieval, *Information Processing and Management,* **15** (5): 247–260, 1979.
30. D. A. Buell and D. H. Kraft, A model for a weighted retrieval system, *J. Amer. Soc. Information Sci.,* **32** (3): 211–216, 1981.
31. A. Bookstein, Fuzzy requests: An approach to weighted Boolean searches, *J. Amer. Soc. Information Sci.,* **31** (4): 240–247, 1980.
32. G. Bordogna and G. Pasi, A fuzzy linguistic approach generalizing Boolean information retrieval: A model and its evaluation, *J. Amer. Soc. Information Sci.,* **44** (2): 70–82, 1993.
33. D. Lucarella, Uncertainty in information retrieval: An approach based on fuzzy sets, *Ninth Annual Int. Phoenix Conference on Computers and Communications,* Los Alamitos CA: IEEE Computer Society Press, 1990, pp. 809–814.
34. L. J. Mazlack and L. Wonboo, Identifying the most effective reasoning calculi for a knowledge-based system, *IEEE Trans. Syst. Man. Cybern.,* **23** (5): 404–409, 1993.
35. D. A. Buell, A general model of query processing in information retrieval systems, *Information Processing and Management,* **17** (5): 236–247, 1981.
36. C. V. Negoita, On the notion of relevance in information retrieval, *Kybernetes,* **2** (3): 112–121, 1973.
37. S. Miyamoto, *Fuzzy sets in Information Retrieval and Cluster Analysis.* Boston: Kluwer, 1990.
38. S. Miyamoto, Two approaches for information retrieval through fuzzy associations, *IEEE Trans. Syst. Man. Cybern.,* **19** (1): 123–130, 1989.
39. E. Neuwirth and L. Reisinger, Dissimilarity and distance coefficients in automation-supported thesauri, *Information Systems,* **7** (1): 54–67, 1982.
40. T. Radecki, Mathematical model of information retrieval system based on the concept of fuzzy thesaurus, *Information Processing and Management,* **12** (5): 298–317, 1976.
41. S. Miyamoto and K. Nakayama, Fuzzy information retrieval based on a fuzzy pseudothesaurus. *IEEE Trans. Syst. Man. Cybern.,* **16** (2): 237–243, 1986.

42. K. Nomoto et al., A document retrieval system based on citations using fuzzy graphs, *Fuzzy Sets and Systems,* **38** (2): 191–202, 1990.

43. Y. Ogawa, T. Morita, and K. Kobayashi, A fuzzy document retrieval system using the keyword connection matrix and a learning method, *Fuzzy Sets and Systems,* **39** (2): 163–179, 1991.

44. D. H. Kraft and D. A. Buell, Fuzzy sets and generalized Boolean retrieval systems, *Int. J. Man-Machine Studies,* **19** (1): 45–56, 1983.

45. S. C. Carter and D. H. Kraft, A generalization and clarification of the Waller-Kraft wish-list, *Information Processing and Management,* **25** (1): 15–25, 1989.

46. M. Bartschi, An overview of information retrieval subjects, *Computer,* **18** (5): 67–74, 1985.

47. W. G. Waller and D. H. Kraft, A mathematical model of a weighted Boolean retrieval system, *Information Processing and Management,* **15** (3): 235–245, 1979.

48. R. R. Yager, A note on weighted queries in information retrieval systems, *J. Amer. Soc. Information Sci.,* **38** (1): 47–51, 1987.

49. P. B. Kantor, The logic of weighted queries, *IEEE Trans. Syst. Man. Cybern.,* **11** (12): 151–167, 1981.

50. G. Chen, E. Kerre, and J. Vandenbulcke, A computational algorithm for the FFD transitive closure and a complete axiomatization of fuzzy functional dependency, *Int. J. of Intelligent Systems,* **9** (3): 421–440, 1994.

51. P. Saxena and B. Tyagi, Fuzzy functional dependencies and independencies in extended fuzzy relational database models, *Fuzzy Sets and Systems,* **69** (1): 65–89, 1995.

52. P. Bosc and O. Pivert, SQLf: A relational database language for fuzzy querying, *IEEE Trans. Fuzzy Syst.,* **3** (1): 1–17, 1995.

53. J. Kacprzyk and S. Zadrozny, FQUERY for ACCESS: Fuzzy querying for Windows-based DBMS, in P. Bosc and J. Kacprzyk (eds.) *Fuzziness in Database Management Systems,* Heidelberg GR: Physica-Verlag, 1995, pp. 415–435.

54. R. George et al., Uncertainty management issues in the object-oriented data model, *IEEE Trans. Fuzzy Syst.,* **4** (2): 179–192, 1996.

55. V. Cross, R. DeCaluwe, and N. VanGyseghem, A perspective from the Fuzzy Object Data Management Group, *Proc. 6th Int. Conf. on Fuzzy Systems,* Los Alamitos, CA: IEEE Computer Society Press, 1997, pp. 721–728.

56. V. Cross, Fuzzy information retrieval, *J. Intelligent Information Syst.,* **11** (3): 115–123, 1994.

FREDERICK E. PETRY
Tulane University