Modeling the Retrieval Process for an Information Retrieval System Using an Ordinal Fuzzy Linguistic Approach

E. Herrera-Viedma
Dept. of Computer Science and A. I., Facultad de Biblioteconomía y Documentación, University of Granada, 18071-Granada, Spain. E-mail: viedma@decsai.ugr.es

A linguistic model for an Information Retrieval System (IRS) defined using an ordinal fuzzy linguistic approach is proposed. The ordinal fuzzy linguistic approach is presented, and its use for modeling the imprecision and subjectivity that appear in the user-IRS interaction is studied. The user queries and IRS responses are modeled linguistically using the concept of fuzzy linguistic variables. The system accepts Boolean queries whose terms can be weighted simultaneously by means of ordinal linguistic values according to three possible semantics: a symmetrical threshold semantic, a quantitative semantic, and an importance semantic. The first one identifies a new threshold semantic used to express qualitative restrictions on the documents retrieved for a given term. It is monotone increasing in index term weight for the threshold values that are on the right of the mid-value, and decreasing for the threshold values that are on the left of the mid-value. The second one is a new semantic proposal introduced to express quantitative restrictions on the documents retrieved for a term, i.e., restrictions on the number of documents that must be retrieved containing that term. The last one is the usual semantic of relative importance that has an effect when the term is in a Boolean expression. A bottom-up evaluation mechanism of queries is presented that coherently integrates the use of the three semantics and satisfies the separability property. The advantage of this IRS with respect to others is that users can express linguistically different semantic restrictions on the desired documents simultaneously, incorporating more flexibility in the user–IRS interaction.

Introduction

Information retrieval involves the development of computer systems for the storage and retrieval of (predominantly) textual information (documents). The main activity of an IRS is the gathering of the pertinent filed documents that best satisfy user information requirements (queries). Both documents and user queries must be formally represented in a consistent way, so that IRS can satisfactorily develop the retrieval activity. Basically, IRSs present three components to carry out their activity:

1. *A Database:* which stores the documents and the representation of their information contents (index terms). It is built using tools for extracting index terms and for representing the documents.
2. *A Query Subsystem:* which allows users to formulate their queries by means of a query language.
3. *An Evaluation Subsystem:* which evaluates the documents for a user query. It presents an inference procedure that establishes a relationship between the user request and the documents in the database to determine the relevance of each document to the user query.

Most of the existing IRSs are based on the Boolean retrieval model (Salton & McGill, 1984; Van Rijsbergen, 1979). Usually, the database represents the documents as sets of index terms, the query subsystem represents the user queries as Boolean combinations of index terms, and the evaluation subsystem uses a total matching mechanism between documents and queries as an inference procedure. These IRSs present many limitations (Cooper, 1988; Salton & McGill, 1984), mainly the lack of flexibility and precision for representing document contents, for describing user queries and for characterizing the relevance of the documents retrieved for a given user query. These drawbacks may be overcome by incorporating weights in the three levels of information representation that exist in an IRS:

1. *Document representation level.* By computing weights of index terms, the system specifies to what extent a document matches the concept expressed by the index terms.
2. *Query representation level.* By attaching weights in a query, a user can provide a more precise description of his or her information needs or desired documents.
3. *Evaluation representation level.* By assigning weights to characterize the relationships between user queries and

document representations the evaluation subsystem provides a means, called retrieval status value (RSV) of a document, to discriminate the documents retrieved by relevance judgments.

The Fuzzy Set Theory (Zadeh, 1965) has been used in order to achieve a mathematical formalization of the use of weights for handling uncertain information in the different representation levels, e.g. (Bookstein, 1980, 1985; Bordogna, Carrara, & Pasi, 1991; Bordogna & Pasi, 1995b; Buell & Kraft, 1981a, 1981b; Cater & Kraft, 1989; Chen & Wang, 1995; Cross, 1994; Kantor, 1981; Kerre, Zenner, & DeCaluwe, 1986; Kraft & Buell, 1983; Lucarella & Morara, 1991; Miyamoto, 1990; Negoita, 1973; Radecki, 1979; Salton, Fox, & Wu, 1983; Tahani, 1976; Waller & Kraft, 1979). These fuzzy retrieval models use predominantly numeric weights (values in [0, 1]) in their fuzzy formulations of the representation levels.

The weights of index terms are usually obtained using automatic full-text indexing procedures without user–IRS interaction (Salton, 1989). Therefore, it seems reasonable to use quantitative values in the representation of document content. However, the other representation levels support the user–IRS interaction, and therefore, they should be able to account the possibility for using qualitative values typical of human communication. The query languages based on numeric weights force the user to quantify qualitative concepts (such as "importance"), ignoring the fact that many users are not able to provide their information needs precisely in a quantitative form but in a qualitative one. In fact, it seems more natural to characterize the contents of the desired documents by explicitly associating a linguistic descriptor to a term in a query, such as "important" or "very important," instead of a numerical value. Similarly, the IRS is more user-friendly if the estimated relevance levels of the documents are supplied in a linguistic form (e.g., linguistic terms such as "relevant," "very relevant," may be used) rather than with scores. Following these ideas (Bordogna & Pasi, 1993), several fuzzy linguistic IRSs have been proposed using a *fuzzy linguistic approach* (Zadeh, 1975) to model the weights in the query and evaluation subsystems (Biswas, Bezdek, Subramanian, & Marques, 1987a, 1987b; Bolc, Kowalski, & Kozlowska, 1985; Bordogna & Pasi, 1993, 1995a; Bordogna & Pasi, 1997; Doszkocs, 1986; Kraft, Bordogna, & Pasi, 1994). In this context, the *ordinal fuzzy linguistic approach* (Delgado, Verdegay, & Vila, 1993; Herrera & Herrera-Viedma, 1997; Herrera, Herrera-Viedma, & Verdegay, 1996b) is a linguistic approach that allows us to overcome the limitations of the classical fuzzy linguistic approach (Zadeh, 1975), i.e., we do not have to explicitly establish semantic rules or syntactic rules (e.g., using a context-free grammar), thereby reducing, the complexity of the design for the IRS.

To formalize fuzzy linguistic weighted querying, we have to arrange the query elements that a user can weigh and some aspects of the semantics associated to the query weights as well. Obviously, a user can weigh three elements in a query: the individual terms of the query, the logical connectives for the query, and the subexpressions for the query (associations of terms with logical connectives). The first option is the one most often applied by users. On the other hand, three semantic possibilities are to be found in the literature (Bordogna et al., 1991; Kraft et al., 1994): weights as measures of the importance of a specific element in representing the query, or as a threshold to aid in matching a specific document to the query, or as a description of an ideal or perfect document. These semantics act on the quality, hence, a term represents the conceptual content of a document. That is to say, usual query subsystems manage qualitative semantics considering that users do not need semantics of a quantitative nature. Thus, a user cannot express his/her possible quantitative restrictions in a query (e.g., to establish limits on the amount of documents to be retrieved for each term). Furthermore, these usual query subsystems manage in a same weighted query only one semantic possibility, and so, they do not support those users that may need to express different kinds of semantic restrictions in a same weighted query.

The main aim of this article is to present a linguistic IRS with a highly expressive weighted query subsystem. It is modeled by means of an ordinal fuzzy linguistic approach to simplify its design. The query subsystem is Boolean and presents two novelties: (i) users can express qualitative or quantitative restrictions on the query terms; and (ii) users can express different kinds of semantic restrictions on a term in a weighted query simultaneously. To do so, we introduce two new semantics, a qualitative one, called the *symmetrical threshold semantic*, and a quantitative other, called the *quantitative semantic.* The first one is modeled by a linguistic matching function that is different from the usual functions proposed in the literature for threshold semantics (monotone nondecreasing function) because it is symmetrical with respect to the mid value, i.e., the function is monotone increasing for the threshold values that are on the right of the mid-threshold value (presence weights), and decreasing for the values that are on the left (absence weights). The latter is modeled by a linguistic matching function, which limits the amount of documents to be retrieved for a term in a query. We also incorporate the usual semantic of relative importance (Waller & Kraft, 1979), but its effect is restricted when the term appears in a Boolean expression. It is modeled by two aggregation operators of weighted linguistic information used to manage the Boolean connectives of the subexpressions: *the Linguistic Weighted Disjunction (LWD) operator*, and the *Linguistic Weighted Conjunction (LWC) operator* (Herrera & Herrera-Viedma, 1997). We define a new weighted query language that increases the expression possibilities for the users. It supports the fact that a user can use all three kinds of semantics on the terms for a query simultaneously or independently. Thus, we incorporate more flexibility in the user–IRS interaction by providing more means for each user to express his/her information needs. The linguistic IRS has a bottom-up evaluation subsystem that deals coherently with the

different semantics that may appear in a weighted query. Its main property is that it acts by overcoming the problems of the application of the importance semantic, i.e., it satisfies the *separability property.* Finally, we should point out that the retrieved documents are arranged in linguistic relevance classes, as was done previously (Bordogna & Pasi, 1993), but in this case identified by ordinal linguistic values.

This article is set out as follows. The ordinal fuzzy linguistic approach is presented next. The fuzzy linguistic IRS is defined in its own section. Finally, the last section includes our conclusions.

The Ordinal Fuzzy Linguistic Approach

There are situations in which the information cannot be assessed precisely in a quantitative form, but it may be done in a qualitative one, and thus, the use of a *linguistic approach* is necessary. For example, when attempting to qualify phenomena related to human perception, we are often led to use words in natural language instead of numerical values. This may arise for different reasons (Chen & Hwang, 1992): there are some situations in which the information may be unquantifiable due to its nature, and thus, it may be stated only in linguistic terms (e.g., when evaluating the "comfort" or "design" of a car, terms like "good," "medium," "bad" can be used). In other cases, precise quantitative information may not be stated because either it is unavailable or the cost of its computation is too high, so an "approximate value" may be tolerated (e.g., when evaluating the speed of a car, linguistic terms like "fast," "very fast," "slow" may be used instead of numerical values).

The *fuzzy linguistic approach* is an approximate technique appropriate for dealing with qualitative aspects of problems. It models linguistic values by means of *linguistic variables* (Zadeh, 1975). Because words are less precise than numbers, the concept of a linguistic variable serves the purpose of providing a measure for an approximate characterization of the phenomena that are too complex or ill-defined to be amenable to their description by conventional quantitative terms. Its application is beneficial because it introduces a more flexible framework for representing the information in a more direct and suitable way when it is not possible to express it accurately. Thus, the burden of quantifying a qualitative concept is eliminated, and the systems can be simplified.

Definition 1 (Zadeh, 1975): A linguistic variable is characterized by a quintuple $(L, H(L), U, G, M)$ in which L is the name of the variable; $H(L)$ (or simply H) denotes the term set of L, i.e., the set of names of linguistic values of L, with each value being a fuzzy variable denoted generically by X and ranging across a universe of discourse U, which is associated with the base variable u; G is a syntactic rule (*which usually takes the form of a grammar*) for generating the names of values of L; and M is a semantic rule for associating its meaning with each L, $M(X)$, which is a fuzzy subset of U.

In any fuzzy linguistic approach for solving a particular problem, we have to make two decisions (Herrera & Herrera-Viedma, 2000):

1. *The choice of the linguistic term set and its semantic.* It consists of establishing the linguistic expression domain used to provide the linguistic performance values. To do so, we have to choose the granularity of the linguistic term set, its labels, and its semantic.
2. *The choice of the aggregation operator of linguistic information.* It consists of establishing an appropriate aggregation operator of linguistic information for aggregating and combining the linguistic performance values.

The Choice of the Linguistic Term Set and its Semantic

The choice of the linguistic term set and its semantic to represent the linguistic information is the first goal to be satisfied in any linguistic approach for solving a particular problem. From a practical point of view, we can find two possibilities to choose the appropriate linguistic descriptors of the term set and their semantic:

1. *The classical fuzzy linguistic approach.* This first possibility defines the linguistic term set by means of a context free grammar, and the semantic of linguistic terms is represented by fuzzy numbers described by membership functions based on certain parameters and on a semantic rule (Bordogna & Pasi, 1993; Kraft et al., 1994; Zadeh, 1975).
2. *The ordinal fuzzy linguistic approach.* The latter defines the linguistic term set by means of an ordered structure of linguistic terms, and the semantic of linguistic terms is derived from their own ordered structure, which may be either symmetrically distributed or not on the interval [0, 1] (Delgado et al., 1993; Herrera & Herrera-Viedma, 1997; Torra, 1996; Yager, 1995).

In both cases, to establish the linguistic descriptors of a linguistic variable, one important aspect to be analyzed is the *granularity of uncertainty*, i.e., the level of discrimination among different counts for uncertainty, in other words, the cardinality of the linguistic term set used to express the linguistic information. This cardinality must be small enough so as not to impose useless precision on the users, and it must be rich enough to allow a discrimination of the assessments in a limited number of degrees. Typical values of cardinality, used in the linguistic models, are odd values, such as 7 or 9, with an upper limit of granularity of 11, or no more than 13, where the mid-term represents an assessment of "approximately 0.5," and the rest of the terms being placed symmetrically around it (Bonissone & Decker, 1986). These classical cardinality values seems to fall in line with Miller's observation about the fact that human beings can reasonably manage to bear in mind seven or so items (Miller, 1956). In the classical fuzzy linguistic approach, the granularity of uncertainty is not easily under control because the grammar may generate a large list of descriptors,

FIG. 1. A symmetrically distributed ordered set of seven linguistic terms.

and we can find inadequate values of cardinalities (very high ones). However, in the ordinal fuzzy linguistic approach, we can control this aspect and provide users with a few but meaningful and useful linguistic descriptors.

As was mentioned earlier, in this article we will assume the ordinal fuzzy linguistic approach, and therefore, we will reduce the complexity when defining a grammar and a semantic rule, and we will be able to explicitly control the granularity of uncertainty.

In the following subsection we characterize the linguistic term set used to represent the linguistic information.

Characterization of the ordinal linguistic term set

When the linguistic term set is defined by means of a grammar, we mainly have to give the primary terms, the modifiers for them, the production rules, the semantic membership functions of the primary terms, and the action semantic rules for the modifiers (Bonissone, 1982; Bordogna & Pasi, 1993; Herrera & Herrera-Viedma, 2000). In an ordinal characterization of the linguistic term set all terms are assumed to be primary ones, and distributed on a scale ([0, 1]) on which a total order is defined (Herrera & Herrera-Viedma, 1997; Torra, 1996; Yager, 1995). In this case, the semantic is introduced from the structure defined on the linguistic term set.

Let $S = \{s_i\}$, $i \in \{0, \ldots, \mathcal{T}\}$ be a finite and totally ordered label set in the usual sense and with odd cardinality as in (Bonissone & Decker, 1986). Depending on the distribution of the linguistic terms on a scale ([0, 1]), there are two possibilities for defining the semantic of the linguistic term set:

1. *Symmetrically distributed terms.* In this case, ordered linguistic term sets that are distributed on a scale, as was mentioned above, with an odd cardinal and the mid-term representing an assessment of "approximately 0.5" and the rest of the terms that are placed symmetrically around it are assumed. The semantic of the linguistic term set is established from the ordered structure of the term set by considering that each linguistic term for the pair $(s_i, s_{\mathcal{T}-i})$ is equally informative. This proposal may be explicitly defined by assigning a subdomain of the reference domain [0, 1] to each linguistic term. For example, a set of seven terms S defined as

   $S = \{s_0 = none, s_1 = very\ low, s_2 = low, s_3 = medium,$
   $s_4 = high, s_5 = very\ high, s_6 = perfect\},$

   in which $s_a < s_b$ iff $a < b$, can be distributed on [0, 1], as is shown in Figure 1.

   In this example, the distribution is a partition of the [0, 1] interval (Bordogna & Pasi, 1997; Yager, 1995). Another possibility for defining the subdomains of each term consists of assigning fuzzy sets to each term (see Herrera et al., 1996b).
2. *Nonsymmetrically distributed terms.* In this case, it is assumed that a subdomain of the reference domain may be more informative than the rest of the domain (Torra, 1996). In such a case, the density of linguistic labels in that subdomain could be greater than the density in the rest of the reference domain, i.e., the ordered linguistic term set would not be symmetrically distributed. For instance, suppose that we require a temperature control system with a very precise behavior when the temperature is "Low." The linguistic term set for this situation would have a distribution over the reference domain similar to that in Figure 2 (in Fig. 2 AN = almost-nil and QL = quite-low) (Torra, 1996).

Without loss of generality, we will assume the first possibility, i.e., symmetrically distributed terms. Furthermore, we require the following properties: (1) The set is ordered: $s_i \geq s_j$ if $i \geq j$. (2) Negation operator: $\text{Neg}(s_i) = s_j$ such that $j = \mathcal{T} - i$. (3) Maximization operator: $\text{MAX}(s_i, s_j) = s_i$ if $s_i \geq s_j$. (4) Minimization operator: $\text{MIN}(s_i, s_j) = s_i$ if $s_i \leq s_j$.

The subdomains of the terms are given by fuzzy numbers defined on the interval [0, 1], which are described by membership functions. As the linguistic assessments are merely approximate ones given by the users, we can consider that linear trapezoidal membership functions are good enough to capture the vagueness of those linguistic assessments, because obtaining more accurate values may be impossible or unnecessary. This representation is achieved by the 4-tuple $(a_i, b_i, \alpha_i, \beta_i)$ (the first two parameters indicate the interval in which the membership value is 1.0; the third and fourth parameters indicate the left and right widths of the support).

Example 1. For example, we can use the following set of nine labels with each associated semantic and $U = [0, 1]$ (base variable domain) to evaluate the linguistic variables in our fuzzy linguistic IRS (Bonissone & Decker, 1986) as is shown in Figure 3:

$$
\begin{aligned}
T &= Total = (1, 1, 0, 0)\\
EH &= Extremely\_High = (0.98, 0.99, 0.05, 0.01)\\
VH &= Very\_High = (0.78, 0.92, 0.06, 0.05)\\
H &= High = (0.63, 0.80, 0.05, 0.06)\\
M &= Medium = (0.41, 0.58, 0.09, 0.07)\\
L &= Low = (0.22, 0.36, 0.05, 0.06)\\
VL &= Very\_Low = (0.1, 0.18, 0.06, 0.05)\\
EL &= Extremely\_Low = (0.01, 0.02, 0.01, 0.05)\\
N &= None = (0, 0, 0, 0)
\end{aligned}
$$

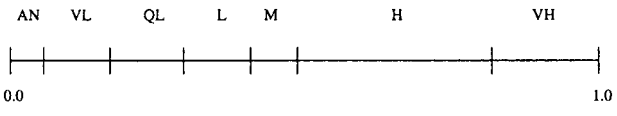

FIG. 2. A nonsymmetrically distributed ordered set of seven linguistic terms.

The Choice of the Aggregation Operator of Linguistic Information

The management of linguistic information requires the use of adequate aggregation operators of linguistic information. One technique to combine linguistic values given on an ordered set of labels like S is the *symbolic computation* (Delgado et al., 1993; Herrera & Herrera-Viedma, 1997; Herrera et al., 1996b; Yager, 1995). It acts by direct computation on labels by taking into account the meaning and features of such linguistic assessments. This symbolic tool seems natural when using the linguistic approach, because the linguistic assessments are simply approximations that are given and handled when it is impossible or unnecessary to obtain more accurate values. Thus, in this case, the use of membership functions associated to the linguistic terms is unnecessary. Furthermore, they are computationally simple and quick (Delgado et al., 1993).

The evaluation subsystem of fuzzy linguistic IRS deals with linguistic weighted queries. Therefore, we need aggregation operators of linguistic weighted information to evaluate the linguistic RVSs of the documents. In the following subsection, we present the operators used in the evaluation subsystem.

Aggregation operators of linguistic weighted information

These operators aggregate linguistic information provided for different criteria that are not equally important. Usually, they provide the aggregation of linguistic weighted information combined with the linguistic importance degrees as a final result as in Herrera and Herrera-Viedma (1997). Therefore, the aggregation of linguistic weighted information involves two activities:

1. The transformation of the linguistic weighted information under the linguistic importance degrees by means of a transformation function h.
2. The aggregation of the transformed linguistic weighted information by means of an aggregation operator of nonweighted linguistic information f.

A general specification of the requirements that any importance transformation function h must satisfy for any type of aggregation operator f is the following (Herrera & Herrera-Viedma, 1997; Yager, 1994): (1) if $a > b$, then $h(c, a) \geq h(c, b)$; (2) $h(c, a)$ is monotone in c; (3) $h(s_0, a) = \text{ID}$; (4) $h(s_{\mathcal{T}}, a) = a$, with $a, b \in S$ expressing the linguistic weighted assessments to be aggregated, $c \in S$ the linguistic importance degree associated with the assessment, and "ID" an identity element, which is such that if we add it to our aggregations it does not change the aggregated value. The first condition means that the function h is monotonically nondecreasing in the second argument, i.e., if the satisfaction with regards to the criteria increases the overall satisfaction should not decrease. The second condition may be viewed as a requirement of the effect of the importance of being consistent. It does not specify whether h is monotonically nonincreasing or nondecreasing in the first argument. It should be noted that conditions (3) and (4) actually determine the type of monotonicity obtained from (2). If $a > \text{ID}$, then $h(c, a)$ is monotonically nondecreasing in c, whereas if $a < \text{ID}$ then it is monotonically nonincreasing. The third condition is a manifestation of the imperative that void importance items do not affect the aggregation process. The final condition is essentially a boundary condition that states that the assumption of each importance equal to one is effectively the same as not including importance at all.

The transformation function depends upon the type of aggregation of weighted information that is going to be performed. In our IRS, we use the linguistic aggregation operators MAX and MIN (see above). In Yager (1987), he discussed the effect of the importance degrees on the "MAX" and "MIN" types of aggregation and suggested a class of functions for importance transformation in both types of aggregation. For the MIN aggregation, he suggested a family of t-conorms acting on the weighted information and the negation of the importance degree, which presents the nonincreasing monotonic property in these importance degrees. For the MAX aggregation, he suggested a family of t-norms acting on weighted information and the importance degree, which presents the nondecreasing monotonic property in these importance degrees. According to these ideas, we propose to use the following aggregation operators of linguistic weighted information (Herrera & Herrera-Viedma, 1997) in the evaluations of the linguistic weighted queries:

Definition 2: The aggregation of a set of linguistic weighted opinions, $\{(c_1, a_1), \ldots, (c_m, a_m,)\}$, $c_i, a_i \in S$, according to the Linguistic Weighted Disjunction (LWD) operator is defined as

$$\text{LWD}[(c_1, a_1), \ldots, (c_m, a_m)] = \text{MAX}_{i=1,\ldots,m}\text{LC}_k^{\rightarrow}(c_i, a_i),$$

where a_i represents the weighted opinion, c_i the importance degree of a_i, and $\text{LC}_k^{\rightarrow}$ are a group of linguistic t-norms, called *the linguistic conjunction functions*, which are monotonically nondecreasing in the weights and satisfy the properties required for any transformation function h. Examples of these functions are:

1. *The classical MIN operator:*
$$\text{LC}_1^{\rightarrow}(c, a) = \text{MIN}(c, a).$$
2. *The nilpotent MIN operator:*
$$\text{LC}_2^{\rightarrow}(c, a) = \begin{cases} \text{MIN}(c, a) & \text{if } c > \text{Neg}(a) \\ s_0 & \text{otherwise.} \end{cases}$$
3. *The weakest conjunction:*
$$\text{LC}_3^{\rightarrow}(c, a) = \begin{cases} \text{MIN}(c, a) & \text{if } \text{MAX}(c, a) = s_{\mathcal{T}} \\ s_0 & \text{otherwise.} \end{cases}$$

Definition 3: The aggregation of a set of linguistic weighted opinions, $\{(c_1, a_1), \ldots, (c_m, a_m)\}$, $c_i a_i \in S$,

according to the Linguistic Weighted Conjunction (LWC) operator is defined as:

$$\mathrm{LWC}[(c_1, a_1), \ldots, (c_m, a_m)] = \mathrm{MIN}_{i=1,\ldots,m}\mathrm{LI}_k^{\rightarrow}(c_i), a_i),$$

where $\mathrm{LI}_k^{\rightarrow}$ are a group of linguistic implication functions, called *the linguistic implication functions*, which are monotonically nonincreasing in the weights and satisfy the properties required for any transformation function h. Examples of these functions are:

1. *Kleene-Dienes's implication function:*
$$\mathrm{LI}_1^{\rightarrow}(c, a) = \mathrm{MAX}(\mathrm{Neg}(c), a).$$
2. *Gödel's implication function:*
$$\mathrm{LI}_2^{\rightarrow}(c, a) = \begin{cases} s_{\mathcal{T}} & \text{if } c \leq a \\ a & \text{otherwise.} \end{cases}$$
3. *Fodor's implication function:*
$$\mathrm{LI}_3^{\rightarrow}(c, a) = \begin{cases} s_{\mathcal{T}} & \text{if } c \leq a \\ \mathrm{MAX}(\mathrm{Neg}(c), a) & \text{otherwise.} \end{cases}$$

Remark 1: The LWD and LWC operators will be used to model the Boolean connectives OR and AND that connects the terms weighted according to the relative importance semantic in the subexpressions for a query, respectively.

The Fuzzy Linguistic IRS

In this section, we present a fuzzy linguistic IRS modeled using an ordinal fuzzy linguistic approach. This linguistic approach is applied in the design of the query subsystem and the evaluation subsystem. Both subsystems provide ordinal linguistic values to express weights of the terms in a query and RSVs of matched documents, respectively. Therefore, the design of the IRS is simplified and the information is presented in a more comprehensible way. The main advantage of this fuzzy linguistic IRS is that users can formulate weighted queries using different semantics, even, simultaneously.

From a mathematical point of view, we define the fuzzy linguistic IRS as a collection of six elements (D, T, Q, F, W, E), where: D is a set of documents or records; T is a set of index terms (single words or phrases); Q is a set of fuzzy linguistic weighted Boolean queries or requests; F is a numeric indexing function that weighs the relationship between D and T with numeric values; W is a linguistic weighting function that weighs the relationship between T and Q with ordinal linguistic values; and E is a linguistic evaluation function that weighs the relationship between Q and D with ordinal linguistic values.

In the following subsections, we define these basic parts of system: the database (D and T and F), the query subsystem (Q and W) and the evaluation subsystem (E).

Definition of the Database

We assume a database of a traditional fuzzy IRS as in (Buell & Kraft, 1981; Miyamoto, 1990; Radecki, 1979; Tahani, 1976; Waller & Kraft, 1979), where the IRS–user interaction is unnecessary because it is built automatically. Therefore, we do not use an ordinal fuzzy linguistic formulation for the database.

The database stores the finite set of documents $D = \{d_1, \ldots, d_m\}$ with its representation $R(D) = \{R_{d_1}, \ldots, R_{d_m}\}$, and the finite set of index terms $T = \{t_1, \ldots, t_l\}$. Documents are represented by means of index terms, which describe the subject content of the documents. A numeric indexing function $F{:}D \times T \rightarrow [0, 1]$, exists, called *index term weighting function* (Tahani, 1976; Waller & Kraft, 1979). Thus, F maps a given document d_j and a given index term t_i to a numeric weight between 0 and 1. $F(d_j, t_i) = 0$ implies that the document d_j is not at all about the concept(s) represented by index term t_i, and $F(d_j, t_i) = 1$ implies that the document d_j is perfectly represented by the concept(s) indicated by t_i. Using the numeric values in (0, 1) F can weigh index terms according to their significance in describing the content of a document in order to improve the retrieval of documents. As is known, the quality of the retrieval results strongly depends on the criteria used to compute F. Different document term weighting schemes have been used for defining F (Bordogna et al., 1991; Cross, 1994; Salton & Buckley, 1988; Salton & McGill, 1984). In this article, we do not focus on this aspect, and assume that the system uses any of the existing weighting methods.

Then, the document representation R_{d_j}, also called *document meaning* (Cross, 1994; Kraft et al., 1994), is viewed as a fuzzy subset of T and characterized by a membership function

$$\mu_{R_{d_j}}{:}T \rightarrow [0, 1], \text{ i.e., } R_{d_j} = \sum_{i=1}^{l} \mu_{R_{d_j}}(t_i)/t_i.$$

$\mu_{R_{dj}}(t_i)$ is a numerical weight that represents the degree of significance of t_i in d_j, such that $\mu_{R_{dj}}(t_i) = F(d_j, t_i)$. In this context, we can define the concept of *meaning of index term* t_i, called $M(t_i)$, which may be represented as a fuzzy subset of documents in D (Buell & Kraft, 1981b; Radecki, 1979):

$$M(t_i) = \sum_{i=1}^{m} \mu_{M(t_i)}(d_j)/d_j,$$

with $\mu_{M(t_i)}(d_j) = F(d_j, t_i) \in [0, 1]$. $M(t_i)$ may be interpreted as the evaluation of a query formed by the single term t_i.

Example 2. Assume a small database that has at this moment a set of 10 index terms $T = \{t_1, \ldots, t_{10}\}$ and a set of seven documents $D = \{d_1, \ldots, d_7\}$. These docu-

ments and are indexed by means of an indexing function F, which assigns the following weights:

$$R_{d_1} = 0.7/t_5 + 0.4/t_6 + 1/t_7$$
$$R_{d_2} = 1/t_4 + 0.6/t_5 + 0.8/t_6 + 0.9/t_7$$
$$R_{d_3} = 0.5/t_2 + 1/t_3 + 0.8/t_4$$
$$R_{d_4} = 0.9/t_4 + 0.5/t_6 + 1/t_7$$
$$R_{d_5} = 0.7/t_3 + 1/t_4 + 0.4/t_5 + 0.8/t_9 + 0.6/t_{10}$$
$$R_{d_6} = 1/t_5 + 0.99/t_6 + 0.8/t_7$$
$$R_{d_7} = 0.8/t_5 + 0.02/t_6 + 0.8/t_7 + 0.9/t_8.$$

Definition of the Query Subsystem

We propose a query subsystem with a fuzzy linguistic weighted Boolean query language to express user queries. With this language each query is expressed as a combination of the weighted index terms that are connected by the logical operators AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). The weights are ordinal linguistic values taken from a label set S. To complete the formulation of the query subsystem we have to study the semantic of the weights and the rules for formulating queries. Both are analyzed in the following subsections.

The semantic of the weights

By assigning weights in queries, users specify restrictions on the documents that the IRS has to satisfy in the retrieval activity. We observe that a user can impose two kinds of restrictions on documents to be retrieved:

1. *Qualitative restrictions:* when the query weights express criteria that affect the quality of the document representations to be retrieved, i.e., constraints to be satisfied by the index term weights that appear in the retrieved document representations.
2. *Quantitative restrictions:* when the query weights express criteria that affect the quantity of the documents to be retrieved, i.e., constraints to be satisfied by the number of documents to be retrieved.

Usually, most classical fuzzy query languages (e.g., see (Biswas et al., 1997a; Bookstein, 1980; Bordogna et al., 1991; Bordogna & Pasi, 1993; Buell & Kraft, 1981a, 1986b; Kraft et al., 1994; Waller & Kraft, 1975)) present these two similarities: (1) they are based on qualitative semantics; and (2) they do not allow users to build weighted queries according to different semantics simultaneously.

However, in some query situations, a user may want to see a *few* documents (quantitative restriction) that concern *very much* (qualitative restriction) with the concept expressed by an index term t_i. To deal with such query situations we propose a more complete and powerful query language that incorporates the following characteristics: (1) it is based on qualitative and quantitative semantics; and (2) it allows users to build weighted queries according to different semantics simultaneously.

In particular, it manages three semantics: two are qualitative, and one is quantitative. We should point out that the chosen semantics are consistent and complementary between one another in the following sense: (i) consistent means that the information needs expressed by some semantics do not contradict those expressed by others; and (ii) complementary means that users can express all or the greater part of their information needs using the chosen semantics. They are presented below.

Qualitative semantics. Fuzzy weights have been used as qualitative restrictions associated to different semantics. The main approaches are the following (Kraft et al., 1994):

1. *Importance semantic* (Bookstein, 1980; Waller & Kraft, 1979). This semantic defines query weights as measures of the relative importance of each term for the query with respect to the others in the query. By associating relative importance weights to terms in a query, the user is asking to see all documents whose content represents the concept that is more associated with the most important terms than with the less important ones. In practice, this means that the user requires that the computation of the RSV of a document be dominated by the more heavily weighted terms.
2. *Threshold semantic* (Buell & Kraft, 1981a, 1981b; Kraft & Buell, 1983). This semantic defines query weights as satisfaction requirements for each term of query to be considered when matching document representations to the query. By associating threshold weights with terms in a query, the user is asking to see all the documents sufficiently about the topics represented by such terms. In practice, this means that the user requires to reward a document whose index term weights F exceed the established thresholds with a high RSV, but allowing some small partial credit for a document whose F values are lower than the thresholds.
3. *Perfection semantic* (Bordogna et al., 1991; Cater & Kraft, 1989). This perfection semantic defines query weights as descriptions of ideal or perfect documents desired by the user. By associating weights with terms in a query, the user is asking to see all the documents whose content satisfies or is more or less close to his ideal information needs as represented in the weighted query. In practice, this means that the user requires to reward a document whose index term weights are equal to or at least near to term weights for a query with the highest RSV. With such a semantic, the user must be able to specify precisely the characteristics of the user's perfect document in a consistent way with the document representations.

In essence, although with different interpretations, the threshold semantic and perfection semantic present many similarities. Hence, most approaches based on both semantics have a similar axiomatic behavior according to the collection of desired properties for the fuzzy IRSs (Cater & Kraft, 1989; Waller & Kraft, 1979), e.g., they usually satisfy

the *separability property.* Both semantics are context free in the sense that a term weight in a query does not carry any information about the relationships between the considered term and the other terms in the query. Furthermore, from a practical point of view, their evaluation mechanisms in the matching processes are usually based on comparison of criteria between index term weights and term weights for the query. So, if we decide to use both semantics at the same time in formulating weighted queries, users may incorporate inconsistencies in their weighted queries. For example, a user may express some information needs by means of a semantic, and at the same time, express the opposite information needs with the other one. To overcome this problem and due to their multiple similarities, in Kraft et al. (1994) a query subsystem was proposed that merged both semantics into one called the *modified threshold semantic.* Its interpretation used a function that merges the evaluation mechanism of the perfection semantic defined in Bordogna et al. (1991) and the evaluation mechanism of the threshold semantic defined in Buell & Kraft (1981a). This function presents the same property as the functions of threshold semantics proposed in the literature, i.e., it is monotone nondecreasing in F. On the other hand, the importance semantic presents many differences with respect to the threshold and perfection semantics. For example, it does not satisfy the separability property, it is not a context-free semantic, and its evaluation mechanism does not depend on comparisons between index term weights and term weights for the query. Furthermore, its semantic interpretation is very different.

Then, from the above analysis, we propose a query language that incorporates the following two qualitative semantics:

1. *A symmetrical threshold semantic*, which presents a different interpretation, being monotone increasing for the threshold values that are on the right of the mid-value, and decreasing for the threshold values that are on the left.
2. *A classical importance semantic*, which has an effect when the term is in a Boolean expression.

We shall present both in detail later.

Quantitative semantic. As was mentioned earlier, a user may want to incorporate in his/her query not only qualitative criteria but also quantitative ones. To model this requirement, some existing systems allow to perform a control on the cardinality of retrieved documents by a whole query (Salton, 1989; Salton & McGill, 1984). In this article, we introduce a new proposal for modeling a semantic of a quantitative nature. This quantitative semantic defines query weights as measures of quantity of documents for each term of query that users want to consider in the computation of the final set of documents retrieved. By associating quantitative weights with the terms in a query, the user is asking to see a set of retrieved documents in which the terms with a greater quantitative weight contribute with a higher number of pertinent documents. In practice, the use of this new quantitative semantic has two beneficial consequences with respect to the classical existing systems:

1. The RSVs of retrieved documents are calculated using the restricted number of document determined for each query term by its quantitative weight. With this weight a user can choose those documents that best satisfy the concepts represented by the term, or most documents that satisfy the concepts, or some documents that satisfy the concepts, etc. Hence, we may perform a refinement or tuning of the output documents of IRS. In our case, this semantic helps us to refine the relevance classes of documents in the output of IRS.
2. A soft control on the total number of retrieved documents that is performed query term to query term.

On the other hand, we must point out that with such a semantic the user must have a clear quantitative idea of the set of retrieved documents for each term that he desires, and in some cases this always is not possible.

To sum up, we propose a query subsystem with a weighted query language which manages three possible semantics: the symmetrical threshold semantic, the importance semantic, and the quantitative semantic.

Rules for formulating queries

Formally, in Bordogna and Pasi (1993) a fuzzy linguistic-weighted Boolean query with one semantic was defined as any legitimate Boolean expression whose atomic components are pairs $\langle t_i, c_i \rangle$ belonging to the set, $T \times H$ (*Importance*); t_i is an element of the set T of terms, and c_i is a value of the linguistic variable, *Importance*, with qualifying the importance that the term t_i must have in the desired documents. The authors proposed a perfection semantic and a classical linguistic approach for defining the linguistic variable *Importance*.

In our case, each term in a query can be weighted according to three different linguistic weights, even simultaneously. As in Bordogna and Pasi (1993), we use the linguistic variable *Importance* to express the linguistic weights, but defining it with the ordinal linguistic approach as described earlier. Thus, we consider a set of ordinal linguistic values S to express the linguistic weights. Then, we define a fuzzy linguistic weighted Boolean query as any legitimate Boolean expression whose atomic components (atoms) are quadruples $\langle t_i, c_i^1, c_i^2, c_i^3 \rangle$ belonging to the set, $T \times S^3$; $t_i \in T$, and c_i^1, c_i^2, c_i^3 are ordinal values of the linguistic variable *Importance*, modeling the symmetrical threshold semantic, the quantitative semantic, and the importance semantic, respectively. Accordingly, the set Q of the legitimate queries is defined by the following syntactic rules:

1. $\forall q = \langle t_i, c_i^1, c_i^2, c_i^3 \rangle \in T \times S^3 \rightarrow q \in Q$.
2. $\forall q, p \in Q \rightarrow q \wedge p \in Q$.

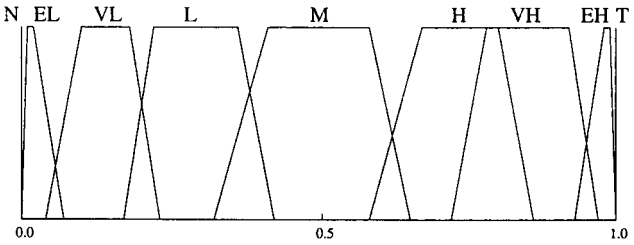

FIG. 3. A set of nine terms with its semantic.

3. $\forall q,\ p \in Q \rightarrow q \vee p \in Q$.
4. $\forall q \in Q \rightarrow \neg(q) \in Q$.
5. All legitimate queries $q \in Q$ are only those obtained by applying rules 1–4, inclusive.

We should point out that all linguistic weights used in a query to model different semantics are primary terms of *Importance*, but with different interpretations depending on their respective semantics. For example, using the set of labels given in Figure 3 $H(Importance) = \{T, EH, VH, H, M, L, VL, EL, N\}$, a query term t_i with a threshold weight of value "*H*" means that the user requires documents in whose content t_i there should be at least one high importance value; however, the same query term t_i with importance weight of value "*H*" means that the user requires that in the computation of the set of retrieved documents the meaning of t_i must have a high importance value.

Remark 2: as in Cater and Kraft (1989), we assume that a term can appear several times in the same query, and therefore, the query subsystem must accept the possibility of having queries with different vectors of three weights on the same terms. To explicitly show this aspect we define a weighting function for queries denoted as *W*. Then, if we fix a query $q_v \in Q$ with $\mathcal{N}$ atoms, *W* is defined as

$$W{:}\{t_v^1, \ldots, t_v^{\mathcal{N}}{:}t_v^k \in T\} \rightarrow S^3$$

$$W(t_v^k) = (w_v^{kh}),\ w_v^{kh} \in S,\ h = 1, 2, 3,$$

where t_v^k is the term of *k*th atom of q_v and w_v^{kh} is the *h*th linguistic weight of t_v^k.

In the following subsection, we explain how to model the matching of weighted queries to documents.

Definition of the Evaluation Subsystem

The goal of evaluation subsystem consists of evaluating documents in terms of their relevance to a weighted query according to three possible semantics. In Cater and Kraft (1989) and Waller and Kraft (1979) some *wish lists* were proposed as a set of properties to be satisfied by evaluation mechanisms of fuzzy-weighted Boolean queries to retain as many of the Boolean lattice properties as possible. Many evaluation subsystems have been designed following these wish lists (Bookstein, 1980; Bordogna et al., 1991; Buell & Kraft, 1981b; Kantor, 1981). These evaluation subsystems evaluate a Boolean query with more than one weighted term by means of a constructive bottom-up process based on the *criterion of separability* (one of the most important properties of the wish list) (Cater & Kraft, 1989; Waller & Kraft, 1979). This process includes two steps:

1. first, the documents are evaluated according to their relevance only to atoms of the query. In this step, a partial RSV is assigned to each document with respect to each atom in the query;
2. second, the documents are evaluated according to their relevance to Boolean combinations of atomic components (their partial RSVs), and so on, working in a bottom-up fashion until the whole query is processed. In this step, a total RSV is assigned to each document with respect to the whole query.

The problem is that the fulfillment of some properties of the wish list may cause conflicts with the preservation of the semantic for the query weights. For example, when using an importance semantic the evaluation of an atom depends on the evaluations of other atoms in the query, and thus, the evaluation process for atoms also depends on the fuzzy connectives chosen to model the Boolean operators. In particular, as happens in Bookstein (1980) and Buell and Kraft (1981), if the AND operator is modeled as a minimum, then it is dominated by the lowest weighted term, and thus this generates inconsistencies with the importance semantic, which requires that lower weighted terms must contribute to a lesser extent to define the total RSV. Furthermore, in such a situation, when modeling the AND operator as a minimum it is impossible to overcome this problem without losing the separability property of the wish list.

In this article, we present a constructive bottom-up evaluation subsystem that satisfies the separability property at the same time as supporting all the semantics of weights considered, even the importance semantic. Its characteristics are analyzed as follows:

1. The RSVs obtained by the evaluation subsystem are linguistic values taken from the linguistic variable "Relevance" as in Bordogna and Pasi (1993), but in this case, it is defined by an ordinal linguistic approach. Therefore, a set of linguistic terms *S* is used to represent the relevance values. For example, if we use the set of labels given in Figure 3, i.e., $H(Relevance) = \{T, EH, VH, H, M, L, VL, EL, N\}$, then a document d_j with a $RSV_j = H$ means that the document presents a high relevance value for the user query processed.
2. As in Bordogna et al. (1991), the evaluation subsystem considers only the terms appearing in the queries. This means that documents are required to be concerned with terms in the queries satisfying the restrictions imposed by the linguistic weights; while, for absent terms, any values are good for the user.
3. To overcome the problems of equivalence in the weighted Boolean queries (Bookstein, 1978; Cater & Kraft, 1989; Waller & Kraft, 1979), the user queries are

preprocessed and put into either a conjunctive normal form (CNF) or a disjunctive normal form (DNF) using the transformation rules given in Korfhage (1978). These rules are applied following the definition of *atomic truth table equivalence* proposed in Cater and Kraft (1989), but assuming the atoms defined to be quadruples $\langle term, weight1, weight2, weight3\rangle$.

4. For a given query, the evaluation subsystem acts as a hierarchical process distinguishing three evaluation levels: (i) evaluation of individual atoms, (ii) evaluation of Boolean subexpressions, and (iii) evaluation of the whole query.
5. The symmetrical threshold and quantitative semantics are applied in the evaluation of individual atoms, because the evaluation process for an atom under such semantics does not depend on the other atoms. Obviously, the separability property is consistent with both semantics.
6. In the evaluation of individual atoms, the symmetrical threshold semantic is applied before the quantitative semantic because the formulation of threshold queries is a more absolute criterion for specifying documents than the use of quantitative weights.
7. For a given query, the evaluation subsystem distinguishes two kinds of logical connectives: (i) weighted logical connectives, which establish relations between the atoms in the subexpressions of a query, and (ii) the nonweighted logical connectives, which establish relations between the subexpressions of a query. For example, in the query $q_1 = (\langle t_1^1, w_1^{11}, w_1^{12}, w_1^{13}\rangle \vee \langle t_1^2, w_1^{21}, w_1^{22}, w_1^{23}\rangle \wedge \langle t_1^3, w_1^{31}, w_1^{32}, w_1^{33}\rangle$, $\vee$ is a weighted logical connective OR and $\wedge$ is a nonweighted logical connective AND.
8. The evaluation subsystem assumes that the importance semantic in a query formed by one atom has no meaning because the importance semantic defines the query weights as measures of the "relative importance" of each atom with respect to the others in the query.
9. Attending to the property (7), the evaluation subsystem imposes that on the normal forms achieved in the preprocessing of queries with more than two atoms, no subexpression can appear with only one atom. For example, if the user provides the following query, $q_1 = (\langle t_1^1, w_1^{11}, w_1^{12}, w_1^{13}\rangle \vee \langle t_1^2, w_1^{21}, w_1^{22}, w_1^{23}\rangle) \wedge \langle t_1^3, w_1^{31}, w_1^{32}, w_1^{33}\rangle$, then the preprocessing mechanism must transform it into $q_1' = (\langle t_{1'}^1, w_{1'}^{11}, w_{1'}^{12}, w_{1'}^{13}\rangle \wedge \langle t_{1'}^3, w_{1'}^{31}, w_{1'}^{32}, w_{1'}^{33}\rangle) \vee (\langle t_{1'}^2, w_{1'}^{21}, w_{1'}^{22}, w_{1'}^{23}\rangle \wedge \langle t_{1'}^3, w_{1'}^{31}, w_{1'}^{32}, w_{1'}^{33}\rangle)$.
10. Due to the interpretation of the importance semantic, in the evaluation subsystem, it is applied in the evaluation of Boolean subexpressions. This is done by integrating the meaning of the importance semantic into the aggregation operators used to model the action of the weighted logical connectives that connect the atoms into Boolean subexpressions of a query. Hence, we manage to keep the independence for the evaluation process of atoms, then, the separability property is satisfied, and as a result, a bottom-up process can be carried out.
11. As was mentioned earlier, the weighted logical connectives AND and OR are modeled by means of the aggregation operators of linguistic weighted information LWC and LWD, respectively. We should note that these operators guarantee the correct application of the importance semantic because both use transformation functions that try to reduce the effect of elements with low importance in the resulting aggregated information. To do so, in the first operator, the elements with low importance are transformed into small values and in the second one into large values (Herrera & Herrera-Viedma, 1997).
12. As queries are preprocessed and put into CNF or DNF form, only atoms in a query are negated. When we have an atom with a negated index term we can negate the weighted term or weigh the negated term. As was done in Buell and Kraft (1981b), the NOT operator is modeled according to the latter interpretation. This means that the evaluation of document d_j for a negated weighted atom $\langle \neg(t_v^k), w_v^{k1}, w_v^{k2}, w_v^{k3}\rangle$ in a query q_v is obtained from the negation of the index term weight, i.e., $1 - F(d_j, t_v^k)$.
13. Finally, the evaluation subsystem models the nonweighted logical connectives AND and OR, which relates Boolean subexpressions into a query, by means of the linguistic functions MIN and MAX, respectively.

Assuming the aforementioned characteristics, the evaluation subsystem evaluates a query in five subsequent steps:

1. Preprocessing of the query.
2. Evaluation of atoms with respect to the symmetrical threshold semantic.
3. Evaluation of atoms with respect to the quantitative semantic.
4. Evaluation of subexpressions and modeling the importance semantic.
5. Evaluation of the whole query.

In the following subsections, we shall study each step in detail.

Preprocessing of the query

In this step, the user query is preprocessed to put it into either CNF or DNF, with the result that all its Boolean subexpressions must have more than two atoms. Weighted single-term queries are kept in their original forms. Then, if we have a query q_v with I subexpressions and $\mathcal{N}$ atoms, it can appear in any one of the forms illustrated graphically in Figure 4, i.e, as AND/Weighted-OR or as OR/Weighted-AND trees.

Evaluation of atoms with respect to the symmetrical threshold semantic

In this step, the documents are evaluated with regard to their relevance to individual atoms in the query, considering only the restrictions imposed by the symmetrical threshold semantic.

Usually, if we have one atom with two components, an index term and a numerical weight $\langle t_i, w_i\rangle$, $t_i \in T$, $w_i \in [0, 1]$, then the evaluation of such an atom is defined as

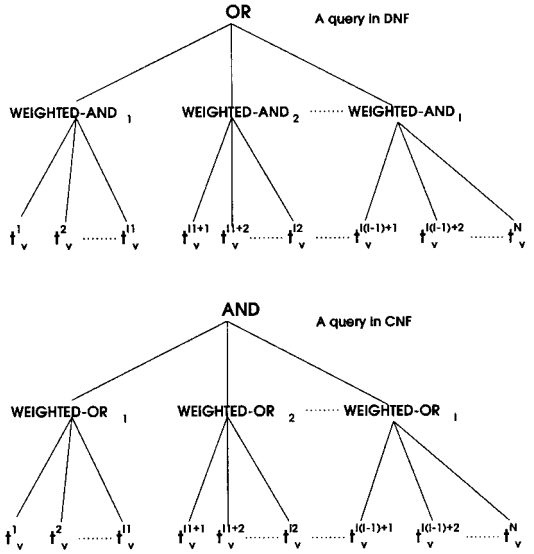

FIG. 4. Queries in normal form.

a fuzzy subset of documents, called $M(t_i)_{w_i}$, obtained from the meaning of index term $M(t_i)$ modified via w_i (Radecki, 1979). $M(t_i)_{w_i}$ is characterized by means of a matching function g:[0, 1] $\times$ [0, 1] $\rightarrow$ [0, 1], which is computed from $F(d_j, t_i)$ and w_i (Kraft et al., 1994; Radecki, 1979), i.e., $\mu_{M(t_i)_{w^i}}(d_j) = g(F(d_j, t_i), w_i), \forall d_j$. $g(F(d_j, t_i), w_i)$ is the evaluation of a document d_j in terms of its relevance with respect to this atom $\langle t_i, w_i \rangle$. In most cases, g is a nondecreasing function in F over the interval [0, 1] (Kraft et al., 1994).

Assuming one atom with four components, an index term and three linguistic weights $\langle t_i, w_i^1, w_i^2, w_i^3 \rangle$, $t_i \in T$, $w_i^h \in S$, then, similarly we define its evaluation with respect to the symmetrical threshold semantic associated to w_i^1 as a fuzzy subset of documents $M(t_i)_{w_i}{}^1$ characterized by means of a linguistic matching function g^1:$S \times S \rightarrow S$,

$$\mu_{M(t_i)_{w_i^1}}(d_j) = g^1(\text{Label}(F(d_j, t_i)), w_i^1), \ \forall d_j,$$

where Label is a function that assigns a label in S to a numeric value $r \in [0, 1]$, defined according to the following expression (Herrera, Herrera-Viedma, & Verdegay, 1996a):

$$\text{Label}(r) = \text{Sup}_q\{s_q \in S : \mu_{s_q}(r) = \text{Sup}_v\{\mu_{s_v}(r)\}\}.$$

To define g^1, we start by analyzing the *modified threshold semantic* for linguistic query weights defined in Kraft et al. (1994). This semantic is the threshold semantic given in Buell and Kraft (1981) redefined by means of the perfection semantic given in Bordogna and Pasi (1993). The linguistic query weights are values of the linguistic variable *Importance*, which is defined by a classical fuzzy linguistic approach. The primary term adopted is *important.* The linguistic weights are variations of the primary term important, e.g., *at least very important, at least averagely important, at least minimally important,* etc. In this model a request $\langle t_i, w_i \rangle$ is synonymous with the query $\langle t_i,$ "*at least* w_i"$\rangle$, and therefore, the request expresses the minimally acceptable documents for a user. Hence, the model assumes that a user always looks for documents with a minimally acceptable presence of a term in their representations. However, in practice, a user does not always search in this way. Such an interpretation is modeled by a numerical nondecreasing matching function g:$[0, 1] \times H(\textit{Importance}) \rightarrow [0, 1]$. To define g, it is necessary determinate a satisfaction range $[p, q] \subseteq [0, 1]$ for each possible linguistic weight. For $F < p$, g measures the closeness of F to range $[p, q]$, for $F \geq q$, g expresses the degree of oversatisfaction of the range $[p, q]$.

We assume that a user can search for documents with a minimally acceptable presence of one term in their representations as in Kraft et al. (1994), or documents with a maximally acceptable absence of one term in their representations. The interpretation adopted for the threshold semantic is the following. When a user asks for documents in which the concept(s) represented by a term t_i is (are) with the value *High Importance,* the user would not reject a document with a F-value greater than *High;* on the contrary, when a user asks for documents in which the concept(s) represented by a term t_i is (are) with the value *Low Importance,* the user would not reject a document with a F-value less than *Low*. Given a request $\langle t_i, w_i^1, —, — \rangle$, this means that the linguistic query weights that imply the presence of a term in a document $w_i^1 \geq s_{\mathcal{T}/2}$ (e.g., *High, Very High,*) it must be treated differently to the linguistic query weights that imply the absence of one term in a document $w_i^1 < s_{\mathcal{T}/2}$ (e.g., *Low, Very Low*). Then, if $w_i^1 \geq s_{\mathcal{T}/2}$ the request $\langle t_i, w_i^1, —, — \rangle$, is synonymous with the request $\langle t_i,$ *at least* $w_i^1, —, — \rangle$, which expresses the fact that the desired documents are those having F-values as high as possible; and if $w_i^1 < s_{\mathcal{T}/2}$ is synonymous with the request $\{t_i,$ at most $w_i^1, —, —\}$, which expresses the fact that the desired documents are those having F-values as low as possible. This interpretation is modeled by the following linguistic matching function g^1:

$$g^1(s_a, s_b) = \begin{cases} s_0 & \text{if } s_b \geq s_{\mathcal{T}/2} \text{ and } s_a = s_0 \\ s_{i_1} & \text{if } s_b \geq s_{\mathcal{T}/2} \text{ and } s_0 < s_a < s_b \\ s_{i_2} & \text{if } s_b \geq s_{\mathcal{T}/2} \text{ and } s_b \leq s_a < s_{\mathcal{T}} \\ s_{\mathcal{T}} & \text{if } s_b \geq s_{\mathcal{T}/2} \text{ and } s_a = s_{\mathcal{T}} \\ s_{\mathcal{T}} & \text{if } s_b < s_{\mathcal{T}/2} \text{ and } s_a = s_0 \\ \text{Neg}(s_{i_1}) & \text{if } s_b < s_{\mathcal{T}/2} \text{ and } s_0 < s_a \leq s_b \\ \text{Neg}(s_{i_2}) & \text{if } s_b < s_{\mathcal{T}/2} \text{ and } s_b < s_a < s_{\mathcal{T}} \\ s_0 & \text{if } s_b < s_{\mathcal{T}/2} \text{ and } s_a = s_{\mathcal{T}} \end{cases}$$

such that

$$i_1 = \text{Max}\left\{0, \text{round}\left(b - \frac{(b-a)}{\mathcal{H}}\right)\right\}$$
$$i_2 = \text{Min}\left\{\mathcal{T}, \text{round}\left(b + \frac{(a-b)}{\mathcal{H}}\right)\right\} \quad \mathcal{H} \in \{1, 2, 3, \ldots, b\}.$$

g^1 is based on the distance or closeness between the linguistic index term weight Label($F(d_j, t_i)$) = s_a and the linguistic query term weight $w_i^1 = s_b$. It can be observed that g^1 is different from the usual matching functions for threshold semantics proposed in the literature (monotone nondecreasing functions) because it is symmetrical with respect to the mid threshold value ($s_{\mathcal{T}/2}$). That is, g^1 is nondecreasing in Label(F) for the threshold values, which are on the right of $s_{\mathcal{T}/2}$, and monotone decreasing in Label(F) for the threshold values that are on the left of $s_{\mathcal{T}/2}$. Therefore, g^1 has opposite behaviors for presence weights and absence weights. When w_i^1 implies presence, then a Label(F) value less than w_i^1 is dealt with as undersatisfying the request (with output s_{i_1}) and a Label(F) value greater than w_i^1 is dealt with as oversatisfying the request (with output s_{i_2}); however, when w_i^1 implies absence, then a Label(F) value less than w_i^1 is dealt with as oversatisfying the request (with output Neg(s_{i_1})) and a Label(F) value greater than w_i^1 must be dealt with as undersatisfying the request (with output Neg(s_{i_2})). Therefore, the matching function must have opposite behaviors for presence weights and absence weights. We note furthermore that the parameter $\mathcal{H}$ is a sensitivity parameter defined to control the importance of the closeness between Label(F) and w_i^1, in the final result. The greater the value of $\mathcal{H}$, the smaller the importance of the value of distance. $\mathcal{H}$ affects the threshold fuzziness, and therefore, different $\mathcal{H}$ values can allow us to model different interpretations of the threshold semantic. When a user indicates that he does not want to impose threshold restrictions on t_i, then g^1 must automatically assume $\mathcal{H} = 1$.

Evaluation of atoms with respect to the quantitative semantic

In this step, the documents continue to be evaluated with regard to their relevance to individual atoms of query, but this time, considering the restrictions imposed by the quantitative semantic.

As mentioned above, a user expresses his/her requirements on the quantity of documents with each term of query that he desires to consider in the computation of the final set of retrieved documents by means of a quantitative semantic. The linguistic quantitative weights are interpreted as follows: when a user establishes a certain number of documents for a term in the query, expressed by a linguistic quantitative weight, then the set of documents to be retrieved must have the minimum number of documents that satisfies the compatibility function representing the meaning of the label represented by the linguistic quantitative weight. Furthermore, these documents must be the ones best represented by the concept(s) implied by the term. In our case, these documents must be those that best satisfy the threshold restrictions imposed on the term.

On the other hand, while the use of a threshold semantic implies the establishment of restrictions on the membership function that characterizes the fuzzy set of documents associated to the meaning of an index term, the use of a quantitative semantic implies the establishment of restrictions on the support of such a fuzzy subset. Therefore, given an atom $\langle t_i, w_i^1, w_i^2, w_i^3\rangle$ and assuming that $M(t_i)_{w_i}{}^1$ is its evaluation according to the symmetric threshold semantic, we model the interpretation of a quantitative semantic by means of a linguistic matching function, called g^2, defined between the support of $M(t_i)_{w_i}{}^1$, called Supp($M(t_i)_{w_i}{}^1$), and the linguistic quantitative weight w_i^2. Then, the evaluation of the atom $\langle t_i, w_i^1, w_i^2, w_i^3\rangle$ with respect to the quantitative semantic associated to w_i^2 is a fuzzy subset of documents $M(t_i)_{w_i}{}^1{}_{,w_i}{}^2$ characterized by the linguistic matching function $g^2{:}\mathcal{P}(D) \times S \times D \rightarrow S$,

$$\mu_{M(t_i)_{w_i^1,w_i^2}}(d_j) = g^2(\text{Supp}(M(t_i)_{w_i^1}), w_i^2, d_j),\ \forall\, d_j$$

$$g^2(\text{Supp}(M(t_i)_{w_i^1}), w_i^2, d_j) = \begin{cases} s_0 & \text{if } d_j \notin B^s \\ \mu_{M(t_i)_{w_i^1}}(d_j) & \text{if } d_j \in B^s \end{cases}$$

where B^s is the set of documents such that $B^s \subseteq$ Supp($M(t_i)_{w_i}{}^1$), obtained according to the following algorithm:

1. $K = \#(\text{Supp}(M(t_i)_{w_i^1}))$.
2. REPEAT
   $M^K = \{s_q \in S{:}\mu_{s_g}(K/m) = \text{Sup}_v\{\mu_{s_v}(K/m)\}\}$.
   $s^K = \text{Sup}_q\{s_q \in M^K\}$.
   $K = K - 1$.
3. UNTIL $((w_i^2 \in M^{K+1})$ OR $(w_i^2 \geq s^{K+1}))$.
4. $B^s = \{d^{\sigma(1)}, \ldots, d^{\sigma(K+1)}\}$, such that $\mu_{M(t_i)_{w_i^1}}(d^{\sigma(h)}) \leq \mu_{M(t_i)_{w_i^1}}(d^{\sigma(l)}) \forall l \leq h$.

According to g^2, the application of the quantitative semantic consists of reducing the number of documents of Supp($M(t_i)_{w_i^1}$), i.e., the number of documents to be considered by the evaluation subsystem for t_i in the later steps. Then, by assigning quantitative weights close to s_0, a user shows his/her preference by considering the most representative document in $M(t_i)_{w_i^1}$ and by assigning quantitative weights close to $s_{\mathcal{T}}$ he does not make a distinction between the documents existing in $M(t_i)_{w_i^1}$. It can be observed that $M(t_i)_{w_i^1,w_i^2} \subseteq M(t_i)_{w_i^1}$ and Supp($M(t_i)_{w_i^1,w_i^2} \subseteq$ Supp($M(t_i)_{w_i^1}$), being $\#(\text{Supp}(M(t_i)_{w_i^1,w_i^2})$ the minimum number of documents that satisfies the linguistic restrictions expressed by the label w_i^2.

We should note that when a user does not want to impose quantitative restrictions on t_i, he must assign a linguistic quantitative weight with a maximum value, i.e., $w_i^2 = s_{\mathcal{T}}$.

Evaluation of subexpressions and modeling the importance semantic

In this step, the documents are evaluated with regards to their relevance to Boolean subexpressions of the queries (Boolean combinations of atoms established by means of the weighted logical connectives), considering the restric-

tions imposed on the connected atoms by the importance semantic.

Let us assume a query q_v with I subexpressions where each subexpression q_v^l ($l = 1, \ldots, I$) has $\#(q_v^l)$ atoms weighted according to the importance semantic with $\#(q_v^l) \geq 2$. From the property 3 given in the subsection *Definition of the Evaluation Subsystem*, we may have two kinds of subexpressions, conjunctive, or disjunctive ones. Therefore, in each subexpression the atoms can be combined by a Weighted-AND or a Weighted-OR, respectively. Due to the interpretation of the importance semantic, which makes sense when the atoms are related to other atoms, the linguistic importance weights are applied in this step of evaluation by means of the linguistic weighted aggregation operators LWC and LWD (Herrera & Herrera-Viedma, 1997), used to model the connectives Weighted-AND and Weighted-OR, respectively. These operators allow us to introduce the importance restrictions by means of their transformation functions and $\mathrm{LI}_h^{\rightarrow}$ and $\mathrm{LC}_h^{\rightarrow}$. Both guarantee that the more important the query terms, the more influential they are in determining the final result. These operators overcome some limitations of classical evaluation mechanisms defined to deal with the importance semantic, e.g., the problems of the AND connective when it is modeled using the fuzzy connective MIN (Bookstein, 1980). In this sense, we should note that in Bordogna & Pasi (1997) a similar proposal was presented, which uses quantifier guided OWA operators (Yager, 1996) to model the logical connectives AND and OR, with the importance semantic being introduced in the computing of the weighting vector that these operators require for their aggregation process.

Then, the evaluation of a subexpression q_v^l, is a fuzzy subset of documents $M(q_v^l)$ characterized a linguistic matching function $g^3{:}(S \times S)^{\#(ql)} \rightarrow S$ defined from the evaluations of its atoms with respect to the symmetrical threshold and quantitative semantics

$$\{M(t_{lv}^k)_{w_{lv}^{k1}, w_{lv}^{k2}},\ k = 1, \ldots, \#(q_v^l)\}$$

and the vector of their respective linguistic importance weights

$$[w_{lv}^{13}, \ldots, w_{lv}^{\#(q_v^l)3}]$$

according to the following expression:

$$\mu_{M(q_v^l)}(d_j) = g^3[(w_{lv}^{k3},\ \mu_{M(t_{lv}^k)w_{lv}^{k1}, w_{lv}^{k2}}),\ k = 1, \ldots, \#(q_v^l)]$$

$\forall d_j$, where if q_v is in DNF then $g^3 =$ LWC and otherwise, $g^3 =$ LWD.

Attending to the definitions of the LWC and LWD operators, when a user does not want to impose importance restrictions on the connected atoms, he must assign the linguistic importance weights s_0 and $s_{\mathcal{T}}$ to all atoms, respectively.

We should point out that from the property 8 (see *Definition of the Evaluation Subsystem*) if $\#(q_v^l) = 1$ then $\mu_{M(q_v^l)}(d_j) = \mu_{M(t_{lv}^1)w_{lv}^{11}, w_{lv}^{12}}(d_j),\ \forall d_j$.

Remark 3: it is observed that in the queries in CNF the interpretation of the importance semantic may produce problems, and for this reason, some authors have suggested allowing only queries in DNF where the importance semantic is applied in the disjuncts of queries. We do not want to simplify the full potentialities of the Boolean query language, and thus, we allow users to formulate queries in both DNF and in CNF. We consider that each atom in a subexpression describes a virtual document set that satisfies the threshold and quantitative restrictions specified on its term, and the importance weight specifies the virtual document set's relative usefulness to the user with respect to the other virtual document sets described by the other atoms.

Evaluation of the whole query

In this final step of evaluation, the documents are evaluated with regards to their relevance to Boolean combinations in all the Boolean subexpressions existing in a query.

Let us assume a query q_v with I subexpressions $\{q_v^1, \ldots, q_v^l\}$, $I \geq 2$. Then, from the property 13 (see *Definition of the Evaluation Subsystem*) the evaluation of q_v is a fuzzy subset of documents $M(q_v)$ characterized by a linguistic matching function $g^4{:}S^I \rightarrow S$ defined from the evaluations of its subexpressions $M(q_v^l)$ according to the following expression:

$$\mu_{M(q_v)}(d_j) = g^4(\mu_{M(q_v^1)}(d_j), \ldots, \mu_{M(q_v^I)}(d_j)),\ \forall d_j$$

where if q_v is in DNF, then $g^4 =$ MAX, and otherwise, $g^4 =$ MIN.

On the other hand, if $I = 1$ then $\mu_{M(q_v)}(d_j) = \mu_{M(q_v^1)}(d_j),\ \forall d_j$.

At the end of this step of evaluation for a query q_v, we find that each document d_j is characterized by a linguistic total $\mathrm{RSV}_j \in S$, such that $\mathrm{RSV}_j = \mu_{M(q_v)}(d_j)$.

Remark 4: we should note that when the evaluation subsystem finishes, the IRS presents the retrieved documents arranged in linguistic relevance classes as in Bordogna and Pasi (1993), but reducing the complexity of the classification process given that the maximal number of classes will be limited by the cardinality of the set of labels chosen for the linguistic variable H(*Relevance*).

Synthesizing the evaluation subsystem by the linguistic evaluation function E

In this subsection, assuming that user queries have been preprocessed and put into the normal form, we synthesize the evaluation subsystem described above using an evaluation function E as in Bordogna et al. (1991), Bordogna and Pasi (1993), and Cross (1994), but defined linguistically on S, i.e., $E{:}Q \times D \rightarrow S$.

After preprocessing a query, we can find the following six kinds of preprocessed queries (q_0, q_1, q_2, q_3, q_4, q_5):

1. $q_0 = \langle t_0^1, w_0^{11}, w_0^{12}, w_0^{13} \rangle \in Q$.
2. $q_1 = \langle \neg t_1^1, w_1^{11}, w_1^{12}, w_1^{13} \rangle \in Q$.
3. $q_2 = \wedge_{k=1}^{\#(q_2 \geq)2} q_v^k \in Q$ such that $q_v^k \in \{q_0, q_1\}$.
4. $q_3 = \vee_{k=1}^{\#(q_3 \geq)2} q_v^k \in Q$ such that $q_v^k \in \{q_0, q_1\}$.
5. $q_4 = \wedge_{l=1}^{I \geq 2} q_3^l \in Q$.
6. $q_5 = \vee_{l=1}^{I \geq 2} q_2^l \in Q$.

Depending on the kind of query, E obtains the RSV_j of any $d_j \in D$ according to the following six rules:

1. $E(q_0, d_j) = g^2(\mathrm{Supp}(M(t_0^1)_{w_0}{}^{11}), w_0^{12}, dj)$, where $\mu_{M(t_0^1)_{w_0}{}^{11}}(d_j = g^1(\mathrm{Label}(F(t_0^1, d_j)), w_0^{11})$.
2. $E(q_1, d_j) = g^2(\mathrm{Supp}(M(\neg t_1^1)_{w_1}{}^{11}), w_1^{12}, dj)$, where $\mu_{M(t_1^1)_{w_1}{}^{11}}(d_j = g^1(\mathrm{Label}(1 - F(t_1^1, d_j)), w_1^{11})$.
3. $E(q_2, d_j) = \mathrm{LWC}[(w_v^{k3}, E(q_v^k, d_j)), k = 1, \ldots, \#(q_2)]$.
4. $E(q_3, d_j) = \mathrm{LWD}[(w_v^{k3}, E(q_v^k, d_j)), k = 1, \ldots, \#(q_3)]$.
5. $E(q_4, d_j) = \mathrm{MIN}(E(q_3^l, d_j), l = 1, \ldots, I)$.
6. $E(q_5, d_j) = \mathrm{MAX}(E(q_2^l, d_j), l = 1, \ldots, I)$.

Example of query evaluation mechanism

In this subsection, we present an example of performance of evaluation subsystem.

Assume the database described in Example 2 and consider the linguistic term set given in Figure 3 to express the values of the linguistic variables H(*Importance*) and H(*Relevance*). Now, consider the following query $q = ((t_5, VH, VL, VH) \wedge (t_6, L, L, VL)) \vee (t_7, H, L, H)$, where the user is declaring his/her interest in a set of documents built, on the one hand, from a very low number of components at least dealing to a very great extent with the concept(s) represented by the term t_5 and a low number of components at most dealing to a much lesser extent with the concept(s) represented by the term t_6, and on the other hand, from a low number of documents dealing to a great extent with the concept(s) represented the by term t_7. Furthermore, the user is indicating that in the evaluation process for the set of desired documents the contribution degree of term t_5 must be more important than the contribution degree for the term t_6, because it is completed with a high contribution degree of the term t_7. From a quantitative perspective the user is declaring his/her interest in a set of documents built using the most representative documents that satisfy the restrictions imposed on each term.

Preprocessing of the query. The query q is in a DNF, but it presents one subexpression with only one atom. Therefore, q must be preprocessed and transformed into a normal form with all its subexpressions with more than two atoms. Then, q is transformed into the following equivalent query $q' = ((t_5, VH, VL, VH) \vee (t_7, H, L, H)) \wedge ((t_6, L, L, VL) \vee (t_7, H, L, H))$, which is expressed in a CNF.

Evaluation of atoms with respect to the symmetrical threshold semantic. First, we obtain the document representation expressed in a linguistic form using the translation function Label:

1. $R_{d_1} = H/t_5 + M/t_6 + T/t_7$.
2. $R_{d_2} = T/t_4 + M/t_5 + H/t_6 + VH/t_7$.
3. $R_{d_3} = M/t_2 + T/t_3 + H/t_4$.
4. $R_{d_4} = VH/t_4 + VL/t_6 + T/t_7$.
5. $R_{d_5} = H/t_3 + T/t_4 + M/t_5 + H/t_9 + M/t_{10}$.
6. $R_{d_6} = T/t_5 + EH/t_6 + H/t_7$.
7. $R_{d_7} = H/t_5 + EL/t_6 + H/t_7 + VH/t_8$.

Let us set the sensitivity parameter $\mathcal{H} = 2$, which gives a large importance to the closeness between linguistic values in g^1. Then, the evaluations of atoms according to the symmetric threshold semantic modeled by g^1 are:

$$M(t_5)_{VH} = VH/d_1 + H/d_2 + H/d_5 + T/d_6 + VH/d_7.$$

$$M(t_7)_H = T/d_1 + VH/d_2 + T/d_4 + H/d_6 + H/d_7.$$

$$M(t_6)_L = M/d_1 + M/d_2 + VH/d_4 + L/d_6 + VH/d_7.$$

Evaluation of atoms with respect to the quantitative semantic. The evaluations of atoms according to the quantitative semantic modeled by g^2 are:

$$M(t_5)_{VH,VL} = T/d_6 \quad .$$

$$M(t_7)_{H,L} = T/d_1 + T/d_4.$$

$$M(t_6)_{L,L} = VH/d_4 + VH/d_7.$$

We note that the quantitative semantic decreases the supports of the evaluations of all atoms. Particularly, the support value in the atom of t_5 that satisfies the restriction imposed by the quantitative weight VL is 1. In the case of t_7 and t_6, the support value that satisfies the restriction of L is 2.

Evaluation of subexpressions and modeling the importance semantic. The query q' has two subexpressions and each one presents two atoms,

$$q'^1 = (t_5, VH, M, VH) \vee (t_7, H, L, H),$$

$$q'^2 = (t_6, L, L, VL) \vee (t_7, H, L, H).$$

Each subexpression is in disjunctive form, and thus, we must use the function $g^3 = \mathrm{LWD}$ to model the Weighted-OR so as to include the effect of the importance semantic in the document evaluation. Fixing the transformation function of LWD as $h = \mathrm{LC}_1^{\rightarrow} = \mathrm{MIN}$, we obtain the following subexpression evaluations:

$$M(q'^1) = H/d_1 + H/d_4 + VH/d_6,$$

$$M(q'^2) = H/d_1 + H/d_4 + L/d_7.$$

For example: $\mu_{M(q'^1)}(d_1) = \text{LWD}[(VH, N), (H, T)] = \text{MAX}\{\text{MIN}(VH, N), \text{MIN}(H, T)\} = H$.

Evaluation of the whole query. Finally, we obtain the document evaluation with respect to the whole query using the matching function $g^4 = \text{MIN}$ to combine the subexpression evaluations: $M(q') = H/d_1 + H/d_4$. Then d_1 and d_4 are displayed to the user in response to the query q, given that they present a high value of relevance.

Remark 5: we should point out that if the quantitative semantic is not considered in the query, then the output would be $M(q') = H/d_1 + H/d_2 + H/d_4 + H/d_6 + H/_7$. Therefore, it is observed that the quantitative semantic has an effect of controlling the number of retrieved documents, and furthermore, refines the class of documents represented by the value H, i.e., it does a tuning of output of IRS. This is a specialy interesting characteristic of the quantitative semantic, overcoat when we deal with database composed by millions document, as for example INTERNET, because by assigning quantitative weights to the query terms a user can obtain a more specific IRS response.

Remark 6: we should point out that in the Boolean expressions based on the AND connective modeled by a linguistic MIN the interpretation of the quantitative semantic may cause problems. In the example, if we consider $(t_5, —, M, —) \wedge (t_6, —, L, —)$, one can expect that the cardinality of the retrieved set of documents to between M and L; however, it is limited by the label L, due to the action of the linguistic operator MIN. We have to say that this semantic is used to impose restrictions on the number of documents that must be considered for each term. It does not impose restrictions on the number of documents that are obtained when the documents retrieved for each term are combined. We think that the problem of the cardinality is related to the operator used to model the logical connective AND. If we use other operators, such as the linguistic OWA operators (Herrera et al., 1996b), this problem may be overcome.

Conclusions

In this article, we have presented a linguistic IRS based on an ordinal fuzzy linguistic approach. With such a linguistic approach we simplify the tasks for designing linguistic IRS based on fuzzy logic, and, for example, we do not have to define syntactic rules to determine the labels. Linguistic modeling has been applied in the representation of the user queries and the IRS responses to improve the user–IRS interaction. The query subsystem accepts Boolean queries with terms weighted by ordinal linguistic values and the evaluation subsystem returns documents arranged in relevance classes labeled with ordinal linguistic values. Its main advantage with respect to others is that users can express both qualitative and quantitative restrictions on the desired documents by means of weights of query terms, and furthermore, these restrictions can be considered simultaneously in the same query. Hence, the system gives users a tool to better specify the characteristics of documents that they desire.

In the design of the query subsystem, we have considered two qualitative semantics, i.e., a new symmetrical threshold semantic, and a usual importance semantic. We have also introduced a new semantic of quantitative nature. With the first one users can express their requirements on the index term weights F for the desired documents by giving minimally acceptable presence values of a term in a document (as usual threshold semantics) or by giving maximally acceptable absence values of a term in a document. The second one is used in a classical way, but it is modeled by means of the aggregation operators of weighted linguistic information; hence, that separability property is preserved, and then a bottom-up evaluation mechanism could be designed. With the third one, users indicate the number of documents for each term that has to be considered in the computing process of a set of desired documents. This last semantic acts by controlling the total number of retrieved documents term to term, but in addition, it performs tuning for the IRS response, as in the example, where the class of documents with the most relevance is refined. With this query subsystem, a user can express a larger number of requirements, but he must decide what and how many semantics must be considered for formulating his/her information needs, the system supports all the possibilities. However, to avoid confusion in its use, the IRS must be completed with a good user interface and a good help system.

In the future, we shall study the use of the semantics in the different weighting levels for queries, as, for example, in the subexpressions or in the connectives.

Acknowledgments

We acknowledge the anonymous referees' comments.

References

Biswas, G., Bezdek, J.C., Subramanian, V., & Marques, M. (1987a). Knowledge-assisted document retrieval: I the natural language interface. Journal of the American Society for Information Science, 38, 50–96.

Biswas, G., Bezdek, J.C., Subramanian, V., & Marques, M. (1987b). Knowledge-assisted document retrieval: II the natural language interface. Journal of the American Society for Information Science, 38, 97–110.

Bolc, L., Kowalski, A., & Kozlowska, M. (1985). A natural language information retrieval system with extensions towards fuzzy reasoning. International Journal of Man–Machine Studies, 23, 335–367.

Bonissone, P.P. (1982). A fuzzy sets based linguistic approach: Theory and applications. In M.M. Gupta & E. Sánchez (Eds.), Approximate reasoning in decision analysis, (pp. 329–339). Amsterdam: North-Holland.

Bonissone, P.P. & Decker, K.S. (1986). Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity. In

L.H. Kanal & J.F. Lemmer (Eds.), Uncertainty in artificial intelligence. (pp. 217–247). Amsterdam: North-Holland.
Bookstein, A. (1978). On the perils of merging Boolean and weighted retrieval systems. Journal of the American Society for Information Science, 29, 156–158.
Bookstein, A. (1980). Fuzzy request: An approach to weighted Boolean searches. Journal of the American Society for Information Science, 31, 240–247.
Bookstein, A. (1985). Probability and fuzzy-set applications to information retrieval. Annual Review of Information Science and Technology, 20, 117–151.
Bordogna, G., Carrara, C., & Pasi, G. (1991). Query term weights as constraints in fuzzy information retrieval. Information Processing & Management, 27, 15–26.
Bordogna, G., & Pasi, G. (1993). A fuzzy linguistic approach generalizing Boolean information retrieval: A model and its evaluation. Journal of the American Society for Information Science, 44, 70–82.
Bordogna, G., & Pasi, G. (1995a). Controlling retrieval through a user-adaptive representation of documents. International Journal of Approximate Reasoning, 12, 317–339.
Bordogna, G., & Pasi, G. (1995b). Linguistic aggretation operators of selection criteria in fuzzy information retrieval. Journal of Intelligent Information Systems, 10, 233–248.
Bordogna, G., & Pasi, G. (1997). Application of the OWA operators to soften information retrieval systems. In R.R. Yager & J. Kacprzyk (Eds), The ordered weighted averaging operators: Theory and applications. (pp. 275–294). Dordrecht: Kluwer Academic Publishers.
Bordogna, G., Fedrizzi, M., & Pasi, G. (1997). A linguistic modelling of consensus in group decision making based on OWA operators. IEEE Transactions on Systems, Man, and Cybernetics, 27, 126–132.
Buell, D., & Kraft, D.H. (1981a). Threshold values and Boolean retrieval systems. Information Processing & Management, 17, 127–136.
Buell, D., & Kraft, D.H. (1981b). A model for a weighted retrieval system. Journal of the American Society for Information Science, 32, 211–216.
Cater, C.S., & Kraft, D.H. (1989). A generalization and clarification of the Waller-Kraft wish list. Information Processing & Management, 25, 15–25.
Chen, S.J., & Hwang, C.L. (1992). Fuzzy multiple attribute decision making—methods and applications. Berlin: Springer-Verlag.
Chen, S.M., & Wang, J.Y. (1995). Document retrieval using knowledge-based fuzzy information retrieval techniques. IEEE Transactions on Systems, Man, and Cybernetics, 25, 793–802.
Cross, V. (1994). Fuzzy information retrieval. Journal of Intelligent Information Systems, 3, 29–56.
Cooper, W. (1988). Getting beyond Boole. Information Processing & Management, 24, 243–248.
Doszkocs, T. (1986). Natural language processing in information retrieval. Journal of the American Society for Information Science, 37, 191–196.
Delgado, M., Verdegay, J.L., & Vila, M.A. (1993). On aggregation operations of linguistic labels. International Journal of Intelligent Systems, 8, 351–370.
Herrera, F., & Herrera-Viedma, E. (1997). Aggregation operators for linguistic weighted information. IEEE Transactions on Systems, Man, and Cybernetics, 27, 646–656.
Herrera, F., & Herrera-Viedma, E. (2000). Linguistic decision analysis: Steps for solving decision problems under linguistic information. Fuzzy Sets and Systems, 115, 67–82.
Herrera, F., Herrera-Viedma, E., & Verdegay, J.L. (1996a). A model of consensus in group decision making under linguistic assessments. Fuzzy Sets and Systems, 78, 73–87.
Herrera, F., Herrera-Viedma, E., & Verdegay, J.L. (1996b). Direct approach processes in group decision making using linguistic OWA operators. Fuzzy Sets and Systems, 79, 175–190.
Kantor, P.B. (1981). The logic of weighted queries. IEEE Transaction on Systems Man and Cybernetics, 11, 816–821.
Kerre, E.E., Zenner, R.B., & DeCaluwe, R.M. (1986). The use of fuzzy set theory in information retrieval and databases: A survey. Journal of the American Society for Information Science, 37, 341–345.
Korfhage, R.R. (1978). Discrete computational structures. New York: Academic Press.
Kraft, D.H., & Buell, D.A. (1983). Fuzzy sets and generalized Boolean retrieval systems. International Journal of Man–Machine Studies, 19, 45–56.
Kraft, D.H., Bordogna, G., & Pasi, G. (1994). An extended fuzzy linguistic approach to generalize Boolean information retrieval. Information Sciences, 2, 119–134.
Lucarella, D., & Morara, R. (1991). FIRST: Fuzzy information retrieval system. Journal of Information Science, 17, 81–91.
Miyamoto, S. (1990). Fuzzy sets in information retrieval and cluster analysis. Boston: Kluwer.
Miller, G.A. (1956). The magical number seven or minus two: Some limits on our capacity of processing information. Psychological Review, 63, 81–97.
Negoita, C.V. (1973). On the application of the fuzzy sets separation theorem for automatic classification in information retrieval system. Information Sciences, 5, 279–286.
Radecki, T. (1979). Fuzzy set theorical approach to document retrieval. Information Processing & Management, 15, 247–260.
Salton, G., Fox, E.A., & Wu, H. (1983). Extended Boolean information retrieval. Communications of the ACM, 26, 1022–1036.
Salton, G., & McGill, M.H. (1984). Introduction to modern information retrieval. New York: McGraw-Hill.
Salton, G., & Buckley, C. (1988). Term weighting approaches in automatic text retrieval. Information Processing & Management, 24, 513–523.
Salton, G. (1989). Automatic text processing: The transformation, analysis of information by computer. Addison Wesley Series in Computer Science. Reading, MA: Addison-Wesley.
Tahani, V. (1976). A fuzzy model of document retrieval systems. Information Processing & Management, 12, 177–187.
Torra, V. (1996). Negation functions based semantics for ordered linguistic labels. International Journal of Intelligent Systems, 11, 975–988.
Van Rijsbergen, C.J. (1979). Information retreival. London: Butterworths.
Waller, W.G., & Kraft, D.H. (1979). A mathematical model of a weighted Boolean retrieval system. Information Processing & Management, 15, 235–245.
Yager, R.R. (1987). A note on weighted queries in information retrieval systems. Journal of the American Society for Information Science, 38, 23–24.
Yager, R.R. (1994). On weighted median aggregation. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2, 101–113.
Yager, R.R. (1995). An approach to ordinal decision making. International Journal of Approximate Reasoning, 12, 237–261.
Yager, R.R. (1996). Quantifier guided aggregation using OWA operators. International Journal of Intelligent Systems, 11, 49–73.
Zadeh, L.A. (1965). Fuzzy sets. Information and Control, 8, 338–353.
Zadeh, L.A. (1975). The concept of a linguistic variable and its applications to approximate reasoning. Part I. Information Sciences, 8, 199–249; Part II. Information Sciences, 8, 301–357; Part III. Information Sciences, 9, 43–80.