Tuning the Matching Function for a Threshold Weighting Semantics in a Linguistic Information Retrieval System

E. Herrera-Viedma,* A.G. López-Herrera,† C. Porcel‡
Department of Computer Science and Artificial Intelligence, Library Science Studies School, University of Granada, 18071, Granada, Spain

Information retrieval is an activity that attempts to produce documents that better fulfill user information needs. To achieve this activity an information retrieval system uses matching functions that specify the degree of relevance of a document with respect to a user query. Assuming linguistic-weighted queries we present a new linguistic matching function for a threshold weighting semantics that is defined using a 2-tuple fuzzy linguistic approach (Herrera F, Martínez L. IEEE Trans Fuzzy Syst 2000;8:746–752). This new 2-tuple linguistic matching function can be interpreted as a tuning of that defined in "Modelling the Retrieval Process for an Information Retrieval System Using an Ordinal Fuzzy Linguistic Approach" (Herrera-Viedma E. J Am Soc Inform Sci Technol 2001;52:460–475). We show that it simplifies the processes of computing in the retrieval activity, avoids the loss of precision in final results, and, consequently, can help to improve the users' satisfaction.

1. INTRODUCTION

The main activity of an information retrieval system (IRS) is the gathering of pertinent archived documents that better satisfy user queries. IRSs present three components to carry out this activity:[1,2]

(1) *A database*, which stores the documents and the representation of their information contents (index terms).
(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 obtaining a retrieval status value (RSV) from each document.

*Author to whom all correspondence should be addressed: e-mail: viedma@decsai.ugr.es.
†e-mail: agabriel@ugr.es.
‡e-mail: cporcel@invest.ugr.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 20, 921–937 (2005)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.20099

The query subsystem supports the user–IRS interaction, and, therefore, it should be able to account for the imprecision and vagueness typical of human communication. This aspect may be modeled by means of the introduction of weights in the query language. Many authors have proposed weighted IRS models using Fuzzy Set Theory.[3–12] Usually, they assume numeric weights associated with the queries (values in $[0, 1]$). However, the use of query languages based on numeric weights forces the user to quantify qualitative concepts (such as "importance"), ignoring 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 desired documents by explicitly associating a linguistic descriptor to a term in a query, like "important" or "very important," instead of a numerical value. In this sense, some fuzzy linguistic IRS models[1,2,13–16] have been proposed using a *fuzzy linguistic approach*[17] to model the query weights and document scores. A useful fuzzy linguistic approach that allows us to reduce the complexity of the design for the IRSs[1,2] is called the *ordinal fuzzy linguistic approach*.[18–21] In this approach, the query weights and document scores are ordered linguistic terms.

On the other hand, we have to establish the semantics associated with the query weights to formalize fuzzy linguistic-weighted querying. There are four semantic possibilities[1,4,14]: weights (i) as a measure of the importance of a specific element in representing the query, (ii) as a threshold to aid in matching a specific document to the query, (iii) as a description of an ideal or perfect document, and (iv) as a limit on the number of documents to be retrieved for a specific element. Usually, in weighted queries, most query subsystems proposed in the literature use only one of the semantic possibilities. In particular, threshold semantics is frequently applied because it is easily understandable by the users.

Assuming an ordinal fuzzy linguistic approach we define a variant for a threshold semantics, called *symmetrical threshold semantics*.[1] This semantics has a symmetric behavior on both sides of the midthreshold value. It assumes that a user may use presence weights or absence weights in the formulation of weighted queries. Then, it is symmetrical with respect to the midthreshold value; that is, it presents the usual behavior for the threshold values that are on the right of the midlinguistic value (presence weights), and the opposite behavior for the values that are on the left (absence weights or presence weights with low value). This semantics means that a user can search for documents with a minimally acceptable presence of one term in their representations or documents with a maximally acceptable absence of one term in their representations. To evaluate this semantics, in Ref. 1 there was defined a parameterized symmetrical linguistic matching function. This function has as its main limitation the loss of precision in final results, that is, in the computation of the linguistic RSVs of documents. The loss of precision appears as a consequence of using a discrete representation for the linguistic terms in the ordinal fuzzy linguistic approach.

In this contribution we present a new modeling of the symmetrical threshold semantics defined in Ref. 1 that overcomes its difficulties. We present a new and alternative definition of the symmetrical matching function that synthesizes the symmetrical threshold semantics and allows us to achieve more precise RSVs, improv-

ing the results of the retrieval and consequently increasing the user's satisfaction. This new symmetrical matching function is defined by using the 2-tuple linguistic representation model,[22] which improves the precision in the representation of linguistic information.

The article is structured as follows. Section 2 presents the preliminaries, that is, the ordinal fuzzy linguistic approach and the 2-tuple fuzzy linguistic representation model together with its operational resources. Section 3 defines the new symmetrical matching function and accomplishes a study of its performance. Section 4 shows an example of the operation of a linguistic IRS with this new symmetrical matching function. Finally, we make some concluding remarks.

2. PRELIMINARIES

In this section, we review some tools of fuzzy linguistic processing that will be used in the new modeling of the symmetrical threshold semantics.

2.1. The Ordinal Fuzzy Linguistic Approach

The *ordinal fuzzy linguistic approach* is an approximate technique appropriate to deal with qualitative aspects of problems.[19,20] An ordinal fuzzy linguistic approach is defined by considering a finite and totally ordered label set $S = \{s_0, \ldots, s_T\}$, $T + 1$ is the cardinality of S in the usual sense, and with odd cardinality (seven or nine labels), with the mid term representing an assessment of "approximately 0.5" and the rest of the terms being placed symmetrically around it.[23] The semantics of the linguistic terms set is established from the ordered structure of the terms set by considering that each linguistic term for the pair (s_i, s_{T-i}) is equally informative. For each label s_i is given a fuzzy number defined on the $[0,1]$ interval, which is described by a membership function. The computational model to combine ordinal linguistic information is based on the following operators:

(1) Negation operator: $Neg(s_i) = s_j, j = T - i$.
(2) Maximization operator: $MAX(s_i, s_j) = s_i$ if $s_i \geq s_j$.
(3) Minimization operator: $MIN(s_i, s_j) = s_i$ if $s_i \leq s_j$.
(4) Aggregation operators: Usually to combine ordinal linguistic information we use aggregation operators based on symbolic computation, for example, the LOWA operator[19] or the LWA operator.[18]

2.2. The 2-Tuple Fuzzy Linguistic Representation Approach

Let $S = \{s_0, \ldots, s_T\}$ be a linguistic term set; if a symbolic method aggregating linguistic information obtains a value $\beta \in [0, T]$ and $\beta \notin \{0, \ldots, T\}$, then an approximation function (app(.)) is used to express the index of the result in S.[22] For example, in the LOWA, app(.) is the simple function *round*.

DEFINITION 1.[22] *Let $\beta \in [0, T]$ be the result of an aggregation of the indexes of a set of labels assessed in a linguistic term set S, that is, the result of a symbolic*

aggregation operation. Let $i = round(\beta)$ and $\alpha_i = \beta - i$ be two values, such that, $i \in \{0,1,\ldots,T\}$ and $\alpha_i \in [-.5,.5)$; then α_i is called a symbolic translation.

From this concept in Ref. 22, Herrera and Martínez developed a linguistic representation model that represents the linguistic information by means of 2-tuples (s_i, α_i), $s_i \in S$ and $\alpha_i \in [-.5,.5)$:

- s_i represents the linguistic label of the information, and
- α_i is a numerical value expressing the value of the translation from the original result β to the closest index label i in S.

This model defines a set of transformation functions between numeric values and linguistic 2-tuples.

DEFINITION 2.[22] *Let S be a linguistic term set and $\beta \in [0, T]$; then the 2-tuple that expresses the information equivalent to β is obtained with the following function:*

$$\Delta : [0, T] \to S \times [-.5,.5)$$

$$\Delta(\beta) = (s_i, \alpha_i), \text{ with } \begin{cases} s_i & i = round(\beta) \\ \alpha_i = \beta - 1 & \alpha_i \in [-.5,.5) \end{cases}$$

where s_i has the closest index label to "β" and "α_i" is the value of the symbolic translation.

PROPOSITION 1.[22] *Let (s_i, α_i), $s_i \in S$ be a linguistic 2-tuple. There is always a Δ^{-1} function, such that, from a 2-tuple it returns its equivalent numerical value $\beta \in [0, T] \subset \Re$.*

Remark 1.[22] From Definition 2 and Proposition 1, it is obvious that the conversion of a linguistic term into a linguistic 2-tuple consists of adding a value 0 as symbolic translation: $s_i \in S \to (s_i, 0)$.

The 2-tuple linguistic computational model operates with the 2-tuples without loss of information and is based on the following operations[22]:

(1) *Negation operator of a 2-tuple:* $Neg(s_i, \alpha_i) = \Delta(T - \Delta^{-1}(s_i, \alpha_i))$.
(2) *Comparison of 2-tuples:* The comparison of linguistic information represented by 2-tuples is carried out according to an ordinary lexicographic order. Let (s_k, α_1) and (s_l, α_2) be two 2-tuples, with each one representing a counting of information:
   - if $k < 1$, then (s_k, α_1) is smaller than (s_1, α_2)
   - if $k = 1$, then
     (1) if $\alpha_1 = \alpha_2$, then (s_k, α_1), (s_1, α_2) represents the same information.
     (2) if $\alpha_1 < \alpha_2$, then (s_k, α_1) is smaller than (s_1, α_2).
     (3) if $\alpha_1 > \alpha_2$, then (s_k, α_1) is bigger than (s_1, α_2).
(3) *Aggregation of 2-tuples:* Using the functions Δ and Δ^{-1}, any numerical aggregation operator can be easily extended for dealing with linguistic 2-tuples. For example, the ordered weighted averaging (OWA)[24] proposed by Yager is an aggregation operator of information that acts taking into account the order of the assessments to be aggregated.

DEFINITION 3.[24] *Let $A = \{a_1, \ldots, a_m\}$, $a_k \in [0,1]$ be a set of assessments to aggregated; then the OWA operator, ϕ, is defined as $\phi(a_1, \ldots, a_m) = W \cdot B^T$, where $W = [w_1, \ldots, w_m]$, is a weighting vector, such that $w_i \in [0,1]$ and $\Sigma_i w_i = 1$, and $B = \{b_1, \ldots, b_m\}$ is a vector associated to A, such that $B = \sigma(A) = \{a_{\sigma(1)}, \ldots, a_{\sigma(m)}\}$, with σ being a permutation over the set of assessments A, such that $a_{\sigma(j)} \leq a_{\sigma(i)}$ $\forall i \leq j$.*

A 2-tuple linguistic extended definition of ϕ would be as follows.

DEFINITION 4. *Let $A = \{(a_1, \alpha_1), \ldots, (a_m, \alpha_m)\}$ be a set of assessments in the linguistic 2-tuple domain; then the 2-tuple linguistic OWA operator, ϕ_{2t}, is defined as $\phi_{2t}((a_1, \alpha_1), \ldots, (a_m, \alpha_m)) = \Delta(W \cdot B^T), B = \sigma(A) = \{(\Delta^{-1}(a_1, \alpha_1))_{\sigma(1)}, \ldots, (\Delta^{-1}(a_m, \alpha_m))_{\sigma(m)}\}$.*

3. A NEW MODELING OF THE SYMMETRICAL THRESHOLD SEMANTICS

In this section we present a new proposal to model the symmetrical threshold semantics defined in Ref. 1 in order to improve its performance. Before presenting it we show the linguistic IRS assumed.

3.1. An Ordinal Linguistic Weighted IRS Based on a Symmetrical Threshold Semantics

In this article, we assume an ordinal linguistic weighted IRS that presents the following elements to carry out its activity:

3.1.1. Database

We assume a database of a traditional fuzzy IRS as in Refs. 7, 10, and 25. The database stores the finite set of documents $\boldsymbol{D} = \{d_1, \ldots, d_m\}$, represented by a finite set of index terms $\boldsymbol{T} = \{t_1, \ldots, t_l\}$, which describe the subject content of the documents. The representation of a document is a fuzzy set of terms characterized by a numeric indexing function F: $\boldsymbol{D} \times \boldsymbol{T} \rightarrow [0, 1]$, which is called the *index term weight*[10]: $d_j = F(d_j, t_1)/t_1 + F(d_j, t_2)/t_2 + \cdots + F(d_j, t_l)/t_l$. F weighs index terms according to their significance in describing the content of a document. Thus $F(d_j, t_i)$ is a numerical weight that represents the degree of significance of t_i in d_j.

3.1.2. Query Subsystem

We use a query subsystem with a fuzzy linguistic-weighted Boolean query language to express user information needs. With this language each query is expressed as a combination of the weighted index terms that are connected by logical operators AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). The weights are ordinal linguistic values taken from a label set S, and they are associated with a symmetrical threshold semantics.[1,2]

Formally, in Ref. 13, a fuzzy linguistic-weighted Boolean query with only one semantics was defined as any legitimate Boolean expression whose atomic components are pairs $\langle t_i, c_i \rangle$, where t_i is an index term and c_i is a value of the linguistic variable, *Importance*, qualifying the importance that the term t_i must have in the desired documents. As in Ref. 13, our atomic components are pairs but defining the linguistic variable *Importance* with the ordinal linguistic approach and associating c_i with a symmetrical threshold semantics. Accordingly, the set $\boldsymbol{Q}$ of the legitimate queries is defined by the following syntactic rules:

(1) $\forall q = \langle t_i, c_i \rangle \in \boldsymbol{T} \times S \rightarrow q \in \boldsymbol{Q}$.
(2) $\forall q, p \in \boldsymbol{Q} \rightarrow q \wedge p \in \boldsymbol{Q}$.
(3) $\forall q, p \in \boldsymbol{Q} \rightarrow q \vee p \in \boldsymbol{Q}$.
(4) $\forall q \in \boldsymbol{Q} \rightarrow \neg q \in \boldsymbol{Q}$.
(5) All legitimate queries $q \in \boldsymbol{Q}$ are only those obtained by applying rules 1–4, inclusive.

3.1.3. Evaluation Subsystem

The evaluation subsystem for weighted Boolean queries acts by means of a constructive bottom-up process based on the *criterion of separability*.[8,10] The RSVs of the documents are ordinal linguistic values whose linguistic components are taken from the linguistic variable *Importance* but representing the concept of *relevance*. Therefore, the set of linguistic terms S is also assumed to represent the relevance values. The evaluation subsystem acts in two steps:

(1) First, the documents are evaluated according to their relevance only to atoms of the query. In this step, the symmetrical threshold semantics is applied in the evaluation of atoms by means of a parameterized linguistic matching function $g : \boldsymbol{D} \times \boldsymbol{T} \times S \rightarrow S$, which is defined as[1]

$$g(d_j, t_i, c_i) = \begin{cases} s_{Min\{a+\beta, T\}} & s_{T/2} \leq s_b \leq s_a \\ s_{Max\{0, a-\beta\}} & s_{T/2} \leq s_b \wedge s_a < s_b \\ Neg(s_{Max\{0, a-\beta\}}) & s_a \leq s_b < s_{T/2} \\ Neg(s_{Min\{a+\beta, T\}}) & s_b < s_{T/2} \wedge s_b < s_a \end{cases}$$

such that (i) $s_b = c_i$; (ii) s_a is the linguistic index weight obtained as $s_a = Label(F(d_j, t_i))$, with $Label : [0,1] \rightarrow S$ being a function that assigns a label in S to a numeric value $r \in [0,1]$; and (iii) β is a bonus value that rewards/penalizes the relevance degrees of documents for the satisfaction/dissatisfaction of request $\langle t_i, c_i \rangle$, which can be defined depending on the closeness between $Label(F(d_j, t_i))$ and c_i, for example, as $\beta = round(2|b - a|/T)$. We should point out that whereas the traditional threshold matching function are always nondecreasing,[14] g is nondecreasing on the right of the midterm and decreasing on the left of the midterm in order to be consistent with the meaning of the symmetrical threshold semantics.

(2) Second, the documents are evaluated according to their relevance to Boolean combinations of atomic components, and so on, working in a bottom-up fashion until the whole query is processed. In this step, the logical connectives AND and OR are modeled by means of LOWA[19] operators with $orness(W) < 0.5$ and $orness(W) \geq 0.5$ respectively, with $orness(W)$ being a orness measure introduced by Yager[24] to classify the aggregation of the OWA operators: $orness(W) = (1/m - 1)(\sum_{i=1}^{m} (m - i) w_i)$.

Remark 2. We should point out that if we have a negated query or a negated subexpression or a negated atom, their evaluation is obtained from the negation of the relevance results computed for the query or the subexpression or atom in a no-negated situation.

3.2. Problems of the Symmetrical Threshold Semantics Modeled by the Parameterized Linguistic Matching Function *g*

According to the symmetrical threshold semantics the evaluation subsystem assumes that users may search for documents with a minimally acceptable presence of one term in their representations (as happens in the classical interpretation[14]) or documents with a maximally acceptable presence of one term in their representations. Then, 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*; conversely, 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 an F value less than *Low*. Given a request $\langle t_i, c_i \rangle \in \boldsymbol{T} \times S$, this means that the query weights that imply the presence of a term in a document $c_i \geq s_{T/2}$ (e.g., *High*, *Very High*) they must be treated differently from the query weights that imply the absence of one term in a document $c_i < s_{T/2}$ (e.g., *Low*, *Very Low*). Then, if $c_i > s_{T/2}$, the request $\langle t_i, c_i \rangle$, is synonymous with the request $\langle t_i, \textit{at least } c_i \rangle$, which expresses the fact that the desired documents are those having F values as high as possible; and $c_i < s_{T/2}$ is synonymous with the request $\langle t_i, \textit{at most } c_i \rangle$, which expresses the fact that the desired documents are those having F values as low as possible.

The linguistic matching function g defined in Ref. 1 represents a possible modeling of the meaning of the symmetrical threshold semantics. However, such modeling or interpretation presents some problems:

(1) *The loss of precision*: This problem is a consequence of the ordinal linguistic framework, which works with discrete linguistic expression domains and this implies assuming limitations in the representation domain of RSVs. Therefore, as linguistic term sets (S) assumed have a limited cardinality (five, seven, or nine labels) to assess the linguistic RSVs, in consequence, it is difficult to distinguish or specify what documents really satisfy better the atomic-weighted request $\langle t_i, c_i \rangle$. Although the system retrieves many documents, the possible relevance assessments are limited by the cardinality of the label set considered.
(2) *The loss of information*: This problem also is a consequence of the ordinal linguistic approach because it forces us to apply approximation operations in the definition of g, in particular, the *rounding* operation used to calculate the parameter β, and as is known,[22] in such a case almost always there exists a loss of information.

Example 1. Let $S = \{s_0 = \textit{Null}\ (N), s_1 = \textit{Extremely\_Low}\ (EL), s_2 = \textit{Very\_Low}\ (VL), s_3 = \textit{Low}\ (L), s_4 = \textit{Medium}\ (M), s_5 = \textit{High}\ (H), s_6 = \textit{Very\_High}\ (VH), s_7 = \textit{Extremely\_High}\ (EH), s_8 = \textit{Total}\ (TO)\}$ be a label set used to assess the linguistic information in an IRS and consider two documents d_1 and d_2, such that $\textit{Label}(F(d_1, t_i)) = EH$ and $\textit{Label}(F(d_2, t_i)) = TO$, respectively; then if the atomic

request is $\langle t_i, M \rangle$, we obtain the same relevance degree for both documents as a consequence of the loss of information, $g(d_1, t_i, M) = TO$ and $g(d_2, t_i, M) = TO$.

(3) *g tends to overvalue the satisfaction/dissatisfaction of the requests*: This problem is a consequence of the definition of g. For example, if we analyze its definition we can observe that relevance degrees generated when the threshold value is satisfied, that is, $s_{Min\{a+\beta,T\}}$ always are limited by the index term weight s_a. This shows a too optimistic evaluation of the satisfaction of the threshold value and reduces the possibilities of discrimination among the documents that satisfy the threshold value. This happens similarly in the dissatisfaction case.

In the following subsection, we try to overcome these problems by defining a new threshold matching function.

3.3. A 2-Tuple Linguistic Matching Function to Model the Symmetrical Threshold Semantics

In this section, we present a new symmetrical matching function to model the symmetrical threshold semantics that overcomes the problems of the matching function g[1] mentioned above. We design it by using as a base the 2-tuple fuzzy linguistic representation model[22] and we call it the 2-tuple linguistic matching function g_{2t}.

First, we should point out that the simple fact of defining the new matching function g_{2t} in a 2-tuple linguistic approach allows us to solve the first problem of g, given that using the 2-tuple linguistic representation model in its definition g_{2t} inherits its properties, and one of the main properties of the 2-tuple linguistic representation model is to eliminate the loss of precision of the ordinal linguistic model.[22]

On the other hand, to overcome the second problem we have to avoid including approximation operations in the definition of g_{2t}, and to overcome the third problem we have to soften the relevance degrees generated by g_{2t} when the threshold value is minimally satisfied by the index term weight.

As mentioned above, symmetrical threshold semantics has a symmetric behavior in both sides of the midthreshold value because it is defined to distinguish two situations in the threshold interpretation: (i) when the threshold value is on the left of the midterm and (ii) when it is on the right. It assumes that a user may use presence weights or absence weights in the formulation of weighted queries. Then, it is symmetrical with respect to the midthreshold value, that is, it presents the usual behavior for the threshold values that are on the right of the midthreshold value (presence weights) and the opposite behavior for the values that are on the left (absence weights or presence weights with low value). Therefore, analyzing the case of presence weights, that is, threshold values that are on the right of the midthreshold value, we rapidly derive the case of absence weights.

When the linguistic threshold weight s_b given by a user is higher, in the usual sense, than middle label of the term linguistic set, $s_{T/2}$, the matching function g is nondecreasing. As we said before, in this case the problem of g is that it rewards excessively those documents whose F values overcome the threshold weight s_b

and penalizes excessively those documents whose F values do not overcome s_b. We look for a nondecreasing matching function g_{2t} that softens the behavior of g. Concretely, to achieve this goal, g_{2t} should work as follows: the more the F values exceed the threshold values and the closer they are to the maximum RSV s_T, the greater the RSVs of the documents. However, when the F values are below the threshold values and closer to s_0, the lower the RSVs of the documents and the closer to s_0 they are. These two circumstances are called in the literature oversatisfaction and undersatisfaction.[14] Assuming a continuous numeric domain $[0, T]$, in Figure 1 we represent graphically the desired behavior of g_{2t} for three possible threshold values $T/2$, u, and u', with values 0, $T/2$, and T being the indexes of the following terms of S: bottom term, middle term, and top term, respectively.

If we focus on the case of threshold value u (see Figure 2), then given two possible values of index term weight $a_1 < u$ and $a_2 > u$, the relevance degrees obtained by a desired matching function should be β_1 and $(T/2) + \beta_2$. Assuming this hypothesis, the definition of the 2-tuple linguistic matching function g_{2t} on the right of the midterm would be as follows:

$$g_{2t} : \boldsymbol{D} \times \boldsymbol{T} \times (S \times [-.5, .5)) \rightarrow (S \times [-.5, .5))$$

$$g_{2t}(d_j, t_i, (s_b, 0)) = \begin{cases} \Delta\left(\beta_2 + \dfrac{T}{2}\right) & \text{if } (s_a, \alpha_a) \geq (s_b, 0) \wedge (s_b, 0) \geq (s_{T/2}, 0) \\ \Delta(\beta_1) & \text{if } (s_a, \alpha_a) < (s_b, 0) \wedge (s_b, 0) \geq (s_{T/2}, 0) \end{cases}$$

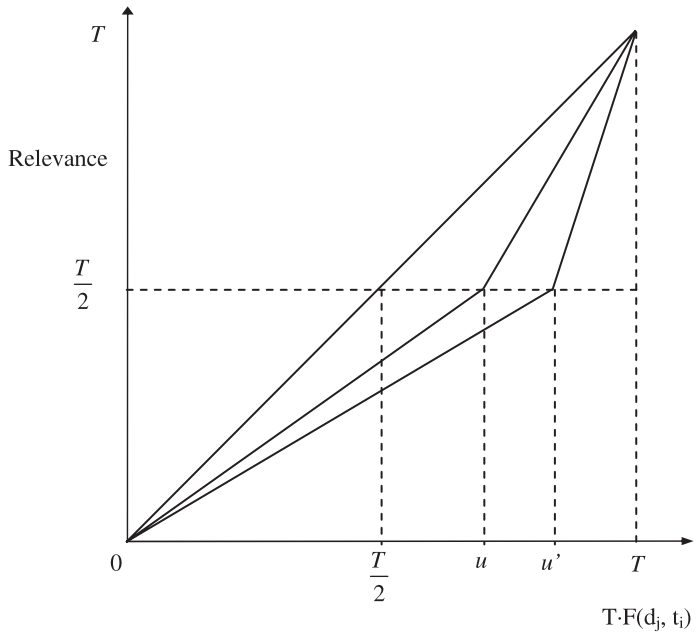

Figure 1. Desired behavior of the matching function g_{2t}.

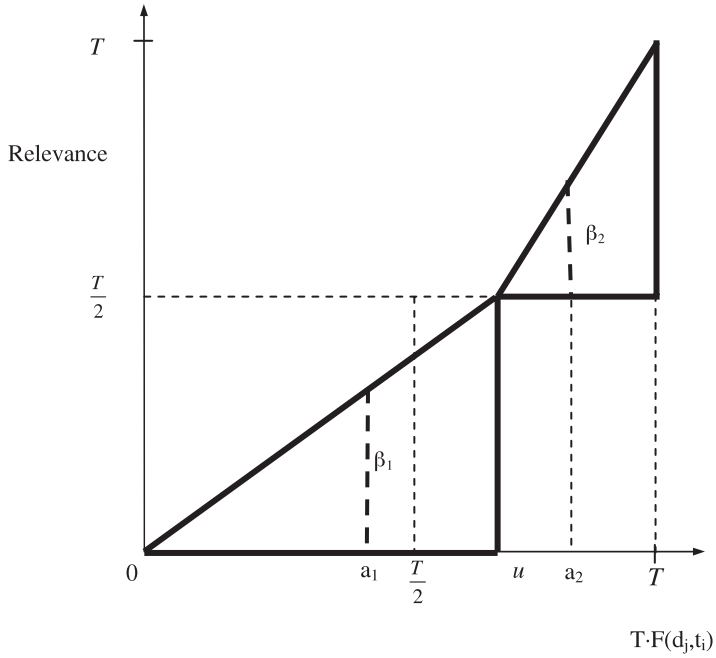

Figure 2. Desired behavior of g_{2t} for a threshold value on the right of the midterm.

where $(s_a, \alpha_a) = \Delta(T \cdot F(d_j, t_i))$, $(s_b, 0)$ is the representation in the linguistic 2-tuple model of the linguistic threshold weight given by a user, and β_1 and β_2 are numerical values obtained as follows. In Figure 2, two triangles are showing the behavior of the desired matching function. The triangle on the right of the midvalue $T/2$ shows the way in which documents that have an index term weight a_2 higher than a threshold value u are rewarded, and the triangle on the left of the midvalue shows the way in which documents that have an index term weight a_1 lower than u are penalized. Analyzing both triangles we can calculate the following expressions for β_2 and β_1:

$$\frac{T - \left(\frac{T}{2}\right)}{T - u} = \frac{\beta_2}{a_2 - u} \Rightarrow \beta_2 = \frac{T \cdot (a_2 - u)}{2 \cdot (T - u)}$$

$$\frac{\frac{T}{2}}{u} = \frac{\beta_1}{a_1} \Rightarrow \beta_1 = \frac{a_1 \cdot \frac{T}{2}}{u} = \frac{a_1 \cdot T}{2 \cdot u}$$

To apply these expressions in the 2-tuple linguistic matching function g_{2t} we must know that:

- $u = \Delta^{-1}(s_b,0)$, with s_b being the linguistic threshold value provided by a user,
- a_2 would be the numeric weight of some index term t_i representing the content of a document d_j, that is, $a_2 = T \cdot F(d_j, t_i)$, and similarly
- a_1 would be the numeric weight of some index term t_i representing the content of a document d_k, that is, $a_1 = T \cdot F(d_k, t_i)$.

Summarizing, given that g_{2t}, like g, must present a symmetric behavior in both sides of the midthreshold value, then the complete definition of g_{2t} is easily obtained as follows:

$$g_{2t}(d_j, t_i, (s_b,0)) = \begin{cases} \Delta\left(\beta_2 + \dfrac{T}{2}\right) & \text{if } (s_a,\alpha_a) \geq (s_b,0) \wedge (s_b,0) \geq (s_{T/2},0) \\ \Delta(\beta_1) & \text{if } (s_a,\alpha_a) < (s_b,0) \wedge (s_b,0) \geq (s_{T/2},0) \\ \Delta\left(\beta_2^* + \dfrac{T}{2}\right) & \text{if } (s_a,\alpha_a) \leq (s_b,0) \wedge (s_b,0) < (s_{T/2},0) \\ \Delta(\beta_1^*) & \text{if } (s_a,\alpha_a) > (s_b,0) \wedge (s_b,0) < (s_{T/2},0) \end{cases}$$

where

$$\beta_2 = \frac{T \cdot (a_2 - u)}{2 \cdot (T - u)}, \qquad \beta_1 = \frac{a_1 \cdot T}{2 \cdot u}, \qquad \beta_2^* = \frac{T \cdot (u - a_1)}{2 \cdot u}, \qquad \beta_1^* = \frac{T \cdot (T - a_2)}{2 \cdot (T - u)}$$

$u = \Delta^{-1}(s_b,0)$, $a_1 = T \cdot F(d_k, t_i)$, and $a_2 = T \cdot F(d_j, t_i)$.

Assuming the label set S defined in Example 1, in Table I we show a comparison of the behavior of both symmetrical matching functions, g and g_{2t} (see the fourth and sixth columns), when $s_b \geq s_{T/2}$, that is, for $s_b \in \{s_4, s_5, s_6, s_7, s_8\}$. To better compare both functions we also show the behavior of the symmetrical matching function g_{2t} projected in an ordinal linguistic domain (see the fifth column), that is considering the results of g_{2t} in the 2-tuple linguistic domain $(S \times 0)$ or ordinal linguistic domain S. Then, analyzing the definition of g_{2t} and the results shown in Table I we can point out the following considerations:

(1) g_{2t} is nondecreasing for threshold values higher than the midthreshold value and decreasing for threshold values lower than the midthreshold value, and therefore, it works like the ordinal linguistic matching function g, being consistent with the meaning of the symmetrical threshold semantics.
(2) The problem of the loss of precision in the results is solved because using the 2-tuple fuzzy linguistic representation model g_{2t} produces more complete and precise results than g, given that relevance results produced express not only the linguistic value obtained in the computing process of the RSVs but also add a numeric measure of the difference of derived information, the so-called symbolic translation.[22] Additionally, we should point out that this improvement in the precision of the results can help to improve the ranking processes of documents in the output of linguistic IRS. For example, in rows 38 and 39 of Table I g returns the same ordinal linguistic RSVs, that is, s_0 and s_0, whereas g_{2t} returns 2-tuple linguistic RSVs $(s_1, -.5)$ and $(s_1, 0)$, respectively. Therefore, in such a case g_{2t} produces more precise results and furthermore, allows us to better rank the documents evaluated in rows 38 and 39.
(3) The problem of the loss of information in the results provided by g_{2t} is also solved because we do not use approximation operations in its definition and the 2-tuple fuzzy

Table I. Comparing linguistic matching functions.

D	$F(d_j, t_i)$	S_b	g	$g_{2t}(S)$	g_{2t}
1	S_0	S_4	S_0	S_0	$(S_0,0)$
2	S_1	S_4	S_0	S_1	$(S_1,0)$
3	S_2	S_4	S_1	S_2	$(S_2,0)$
4	S_3	S_4	S_3	S_3	$(S_3,0)$
5	S_4	S_4	S_4	S_4	$(S_4,0)$
6	S_5	S_4	S_5	S_5	$(S_5,0)$
7	S_6	S_4	S_7	S_6	$(S_6,0)$
8	S_7	S_4	S_8	S_7	$(S_7,0)$
9	S_8	S_4	S_8	S_8	$(S_8,0)$
10	S_0	S_5	S_0	S_0	$(S_0,0)$
11	S_1	S_5	S_0	S_1	$(S_1,-0.2)$
12	S_2	S_5	S_1	S_2	$(S_2,-0.4)$
13	S_3	S_5	S_2	S_2	$(S_2,0.4)$
14	S_4	S_5	S_4	S_3	$(S_3,0.2)$
15	S_5	S_5	S_5	S_4	$(S_4,0)$
16	S_6	S_5	S_6	S_5	$(S_5,0.33)$
17	S_7	S_5	S_8	S_7	$(S_7,-0.33)$
18	S_8	S_5	S_8	S_8	$(S_8,0)$
19	S_0	S_6	S_0	S_0	$(S_0,0)$
20	S_1	S_6	S_0	S_1	$(S_1,-0.33)$
21	S_2	S_6	S_1	S_1	$(S_1,0.33)$
22	S_3	S_6	S_2	S_2	$(S_2,0)$
23	S_4	S_6	S_3	S_3	$(S_3,-0.33)$
24	S_5	S_6	S_5	S_3	$(S_3,0.33)$
25	S_6	S_6	S_6	S_4	$(S_4,0)$
26	S_7	S_6	S_7	S_6	$(S_6,0)$
27	S_8	S_6	S_8	S_8	$(S_8,0)$
28	S_0	S_7	S_0	S_0	$(S_0,0)$
29	S_1	S_7	S_0	S_1	$(S_1,-0.43)$
30	S_2	S_7	S_1	S_1	$(S_1,0.14)$
31	S_3	S_7	S_2	S_2	$(S_2,0.29)$
32	S_4	S_7	S_3	S_2	$(S_2,0.29)$
33	S_5	S_7	S_4	S_3	$(S_3,0.14)$
34	S_6	S_7	S_6	S_3	$(S_3,0.43)$
35	S_7	S_7	S_7	S_4	$(S_4,0)$
36	S_8	S_7	S_8	S_8	$(S_8,0)$
37	S_0	S_8	S_0	S_0	$(S_0,0)$
38	S_1	S_8	S_0	S_1	$(S_1,-0.5)$
39	S_2	S_8	S_0	S_1	$(S_1,0)$
40	S_3	S_8	S_2	S_2	$(S_2,-0.5)$
41	S_4	S_8	S_3	S_2	$(S_2,0)$
42	S_5	S_8	S_4	S_3	$(S_3,-0.5)$
43	S_6	S_8	S_5	S_3	$(S_3,0)$
44	S_7	S	S_7	S_4	$(S_4,-0.5)$
45	S_8	S_8	S_8	S_4	$(S_4,0)$

linguistic representation allows us to gather all information generated in the processes of computing with words carried out by the application of g_{2t}. For example, in Table I we can observe that in many cases (rows 11–14, 20–24, 29–34, 38, 40, 42, 44) if we work with the function g_{2t} in an ordinal linguistic context, there exists a loss of information because the value of symbolic translation is not represented.

(4) With regard to the overvaluation problem of g, we can say that g_{2t} softens that overvaluation behavior of g. For example, if we compare the expressions of both functions in the case of a threshold value on the right of the midlinguistic value and in a satisfaction situation, the results returned by g_{2t} are in the 2-tuple linguistic interval $[(s_{T/2}, 0), (s_T, 0)]$ (using the projection of g_{2t} on an ordinal linguistic domain $S(g_{2t}(S))$), this means they are assessed in the label set

$$\{s_{T/2}, s_{T/2+1}, \ldots, s_T\}$$

while the results returned by g are assessed in the label set

$$\{s_p = Label(F(d_j, t_i)), s_{p+1}, \ldots, s_T\}$$

with $s_p = Label(F(d_j, t_i))$ being the ordinal linguistic weight of the index term t_i representing the content of the document d_j equal to the desired threshold value s_b and maintaining the following relationship:

$$g(d_j, t_i, s_b) \geq g_{2t}(S)(d_j, t_i, s_b) \quad \text{for all } Label(F(d_j, t_i)) \geq s_p \geq s_{T/2}$$

This fact is easily observable in Table I. This happens similarly in the dissatisfaction case.

4. OPERATION OF A LINGUISTIC WEIGHTED IRS BASED ON THE 2-TUPLE LINGUISTIC MATCHING FUNCTION g_{2t}

In this section, we present an example of performance of the IRS defined in Subsection 3.1 under the 2-tuple linguistic symmetrical matching function g_{2t}. This linguistic IRS was defined in an ordinal linguistic context. Then, to show the performance of g_{2t}, that IRS must be redefined in terms of the 2-tuple fuzzy linguistic representation model. To do that, we have to include some modifications in the ordinal linguistic IRS model presented in Subsection 3.1. These modifications affect the evaluation subsystem in particular, keeping the database and query subsystem invariable. They are the following:

- The ordinal linguistic threshold weights of queries provided by the users have to be transformed to the linguistic 2-tuple domain $S \times [-.5, .5)$ to be processed by the evaluation subsystem. As we said in Subsection 2.2, this is carried out by adding the symbolic translation value 0.
- The numeric index term weights $F(d_j, t_i)$ have to be transformed to the 2-tuple linguistic domain, $S \times [-.5, .5)$ by means of the transformation function Δ, as $\Delta(T \cdot F(d_j, t_i))$.
- In the IRS defined in Subsection 3.1 the Boolean connectives of the queries are modeled by means of the LOWA operator. Now, we substitute it by the 2-tuple linguistic OWA operator, ϕ_{2t}, introduced in Definition 4.
- Similarly, in the case of the negated queries, we must substitute the ordinal linguistic negation operator by the 2-tuple linguistic negation operator.

Let us suppose a small database containing a set of seven documents $D = \{d_1, \ldots, d_7\}$, represented by means of a set of 10 index terms $T = \{t_1, \ldots, t_{10}\}$. Documents are indexed by means of an indexing function F, which represents them as follows:

$$d_1 = 0.7/t_5 + 0.4/t_6 + 1/t_7$$

$$d_2 = 1/t_4 + 0.6/t_5 + 0.8/t_6 + 0.9/t_7$$

$$d_3 = 0.5/t_2 + 1/t_3 + 0.8/t_4$$

$$d_4 = 0.9/t_4 + 0.5/t_6 + 1/t_7$$

$$d_5 = 0.7/t_3 + 1/t_4 + 0.4/t_5 + 0.8/t_9 + 0.6/t_{10}$$

$$d_6 = 0.8/t_5 + 0.99/t_6 + 0.8/t_7$$

$$d_7 = 0.8/t_5 + 0.02/t_6 + 0.8/t_7 + 0.9/t_8$$

Using the set of the nine labels given in Example 1 to provide the linguistic weighted queries, consider that a user formulates the following query:

$$q = ((t_5, VH) \vee (t_7, H)) \wedge ((t_6, L) \vee (t_7, H))$$

Then, the evaluation process of this query is developed in the following steps:

(1) *Evaluation of the atoms with respect to the symmetrical threshold semantics*. In this step, first we obtain the documents represented in a 2-tuple linguistic form, applying the function Δ over index term weights $F(d_j, t_i)$:

$$d_1 = (VH, -.4)/t_5 + (L, .2)/t_6 + (TO, 0)/t_7$$

$$d_2 = (TO, 0)/t_4 + (H, -.2)/t_5 + (VH, .4)/t_6 + (EH, .2)/t_7$$

$$d_3 = (M, 0)/t_2 + (TO, 0)/t_3 + (VH, .4)/t_4$$

$$d_4 = (EH, .2)/t_4 + (M, 0)/t_6 + (TO, 0)/t_7$$

$$d_5 = (VH, -.4)/t_3 + (TO, 0)/t_4 + (L, .2)/t_5 + (VH, .4)/t_9 + (H, -.2)/t_{10}$$

$$d_6 = (VH, .4)/t_5 + (TO, -.08)/t_6 + (VH, .4)/t_7$$

$$d_7 = (VH, .4)/t_5 + (N, .16)/t_6 + (VH, .4)/t_7 + (EH, .2)/t_8$$

Then, we evaluate atoms according to the symmetrical threshold semantics by means of g_{2t}:

- (t_5, VH):

$$\{RSV_1^5 = (M, -.27), RSV_2^5 = (L, .2), RSV_5^5 = (VL, .13), RSV_6^5 = (H, -.2),$$

$$RSV_7^5 = (H, -.2)\}$$

- (t_6, L):

$$\{RSV_1^6 = (M, -.16), RSV_2^6 = (EL, .28), RSV_4^6 = (L, .2), RSV_6^6 = (N, .06),$$

$$RSV_7^6 = (TO, -.16)\}$$

- (t_7, H):

$$\{RSV_1^7 = (TO, 0),\ RSV_2^7 = (EH, -.07),\ RSV_4^7 = (TO, 0),\ RSV_6^7 = (VH, -.13),$$

$$RSV_7^7 = (VH, -.13)\}$$

where $RSV_j^i = g_{2t}(d_j, t_i, (c_i, 0))$ and where, for example, the value RSV_2^7 is calculated by means of g_{2t} as follows:

$$RSV_2^7 = g_{2t}(d_2, t_7, (H, 0)) = \Delta\left(\frac{8 \cdot (7.2 - 5)}{2 \cdot (8 - 5)} + \frac{8}{2}\right) = \Delta(6.93) = (s_7 = EH, -.7)$$

(2) *Evaluation of subexpressions*. The query q has two subexpressions, $q_1 = (t_5, VH) \vee (t_7, H)$ and $q_2 = (t_6, L) \vee (t_7, H)$. Each subexpression is in disjunctive form, and thus, we must use an operator ϕ_{2t} with $orness(W) > 0.5$ (for example, with $W = [0.7, 0.3]$) to process them. The results that we obtain are the following:

- $q_1 = (t_5, VH) \vee (t_7, H)$:

$$\{RSV_1^1 = (EH, -.28),\ RSV_2^1 = (VH, -.19),\ RSV_4^1 = (VH, -.4),$$

$$RSV_5^1 = (EL, .49),\ RSV_6^1 = (VH, -.45),\ RSV_7^1 = (VH, -.45)\}$$

- $q_2 = (t_6, L) \vee (t_7, H)$:

$$\{RSV_1^2 = (EH, -.25),\ RSV_2^2 = (H, .24),\ RSV_4^2 = (EH, -.44),$$

$$RSV_6^2 = (M, .13),\ RSV_7^2 = (EH, .25)\}$$

where RSV_j^i is the evaluation result of the subexpression q_i with respect to the document d_j, where, for example, the RSV_2^2 is calculated by means of the 2-tuple linguistic OWA operator ϕ_{2t} as follows:

$$RSV_2^2 = \phi_{2t}(RSV_2^6 = (EL, .28),\ RSV_2^7 = (EH, -.07)) = \Delta(6.93 \cdot 0.7 + 1.28 \cdot 0.3)$$

$$= \Delta(5, 24) = (H, .24)$$

such that $\Delta^{-1}(EL, .28) = 1.28$ and $\Delta^{-1}(EH, -.07) = 6.93$.

(3) *Evaluation of the whole query*. We evaluate the whole query using an operator ϕ_{2t} with $orness(W) < 0.5$ (e.g., with $W = [0.3, 0.7]$) given that it is in a conjunctive normal form, obtaining the following relevance results RSV_j for each document d_j:

$$\{RSV_1 = (EH, -.27),\ RSV_2 = (H, .41),\ RSV_4 = (VH, -.11),\ RSV_5 = (N, .45),$$

$$RSV_6 = (H, -.44),\ RSV_7 = (VH, .06)\}.$$

To evaluate the impact of the 2-tuple linguistic matching function g_{2t} on the performance of IRS, we can compare it with the result obtained by the IRS in an ordinal linguistic framework and applying the linguistic matching function g:

$$\{RSV_1 = EH,\ RSV_2 = VH,\ RSV_4 = VH,\ RSV_5 = EL,\ RSV_6 = H,\ RSV_7 = H\}$$

In analyzing these results, we should point out the following:

(1) First, the advantage of using the 2-tuple fuzzy linguistic representation model is obvious, given that if we use an ordinal linguistic representation it is impossible to distinguish the relevance difference between some documents, for example between d_2 and d_4 or between d_6 and d_7, and these facts are easily observable using the 2-tuple linguistic format.
(2) On the other hand, we must point out that the IRS based on the 2-tuple linguistic matching function g_{2t} obtains results more consistent that better reflect the relevance degree of some documents with respect to the information need expressed by the user. For example:

- If we see the representation of the document d_5, this document does not satisfy any criteria expressed on the weighted query q, that is, it does not contain terms t_6 and t_7, and although it contains the term t_5, its index term weight is lower than the threshold value associated with t_5 in the query, and therefore, it seems more reasonable and consistent to assess this satisfaction situation with a relevance value N (*Null*) than with a value EL (*Extremely_Low*).
- If we see the representation of the documents d_1 and d_7, we can observe that both documents present a very similar satisfaction level with respect to the query; however, the IRS based on g returns for both relevance degrees that are more different than in the case of the IRS based on g_{2t}.

5. CONCLUDING REMARKS

In this article we have described a new modeling of the symmetrical threshold semantics[1] in a linguistic framework. We have defined a new symmetrical linguistic matching function to model the meaning of the symmetrical threshold semantics that overcomes the problems found in the linguistic matching function defined in Ref. 1. We have defined this new linguistic matching function in a 2-tuple fuzzy linguistic context[22] to take advantage of the usefulness of the 2-tuple fuzzy linguistic representation model with respect to avoiding the problems of loss of precision and information in the results.

In the future, we shall research the different threshold matching functions existing in the literature in order to define a general application framework that facilitates their design and use in the IRSs.

References

1. Herrera-Viedma E. Modelling the retrieval process for an information retrieval system using an ordinal fuzzy linguistic approach. J Am Soc Inform Sci Technol 2001;52:460–475.
2. Herrera-Viedma E. An information retrieval system with ordinal linguistic weighted queries based on two weighting elements. Int J Uncertainty Fuzziness Knowl Base Syst 2001;9:77–88.
3. Bookstein A. Fuzzy request: An approach to weighted Boolean searches. J Am Soc Inform Sci 1980;31:240–247.
4. Bordogna G. Carrara C, Pasi G. Query term weights as constraints in fuzzy information retrieval. Inform Process Manag 1991;27:15–26.
5. Bordogna G, Pasi G. Linguistic aggregation operators of selection criteria in fuzzy information retrieval. Int J Intell Syst 1995;10:233–248.

6. Buell D, Kraft DH. Threshold values and Boolean retrieval systems. Inform Process Manag 1981;17:127−136.
7. Buell D, Kraft DH. A model for a weighted retrieval system. J Am Soc Inform Sci 1981; 32:211–216.
8. Cater CS, Kraft DH. A generalization and clarification of the Waller-Kraft wish list. Inform Process Manag 1989;25:15–25.
9. Kraft DH, Buell D.A. Fuzzy sets and generalized Boolean retrieval systems. Int J Man Mach Stud 1983;19:45–56.
10. Waller WG, Kraft DH. A mathematical model of a weighted Boolean retrieval system. Inform Process Manag 1979;15:235–245.
11. Yager RR. A hierarchical document retrieval language. Inform Retriev 2000;3:357–377.
12. Yager RR. A note on weighted queries in information retrieval systems. J Am Soc Inform Sci 1987;38:23–24.
13. Bordogna G, Pasi G. A fuzzy linguistic approach generalizing Boolean information retrieval: A model and its evaluation. J Am Soc Inform Sci 1993;44:70–82.
14. Kraft DH, Bordogna G, Pasi G. An extended fuzzy linguistic approach to generalize Boolean information retrieval. Inform Sci 1994;2:119–134.
15. Herrera-Viedma E, Cordón O, Luque M, Lopez AG, Muñoz AM. A model of fuzzy linguistic IRS based on multi-granular linguistic information. Int J Approx Reas 2003;34:221–239.
16. Bordogna G, Pasi G. An ordinal information retrieval model. Int J Uncertainty Fuzziness Knowl Base Syst 2001;9:63–76.
17. Zadeh LA. The concept of a linguistic variable and its applications to approximate reasoning: Part I. Inform Sci 1975;8:199–249; Part II. Inform Sci 1975;8:301–357; Part III. Inform Sci 1975;9:43–80.
18. Herrera F, Herrera-Viedma E. Aggregation operators for linguistic weighted information. IEEE Trans Syst Man Cybern A: Syst Hum 1997;27:646–656.
19. Herrera F, Herrera-Viedma E, Verdegay JL. Direct approach processes in group decision making using linguistic OWA operators. Fuzzy Set Syst 1996;79:175–190.
20. Herrera F, Herrera-Viedma E, Verdegay JL. A model of consensus in group decision making under linguistic assessments. Fuzzy Set Syst 1996;78:73–87.
21. Yager RR. An approach to ordinal decision making, Int J Approx Reas 1995;12:237–261.
22. Herrera F, Martínez L. A 2-tuple fuzzy linguistic representation model for computing with words. IEEE Trans Fuzzy Syst 2000;8:746–752.
23. Bonissone PP, Decker KS. Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity. In: Kanal LH, Lemmer JF, editors. Uncertainty in artificial intelligence. Amsterdam: North-Holland; 1986. pp 217–247.
24. Yager RR. On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Trans Syst Man Cybern 1988;18:183–190.
25. Miyamoto S. Fuzzy sets in information retrieval and cluster analysis. Norwell, MA: Kluwer Academic Publishers; 1990.