

Weakening of Fuzzy Relational Queries: an Absolute Proximity Relation-based Approach

P. Bosc, A. HadjAli and O. Pivert
IRISA/ENSSAT, Université de Rennes 1
6, rue de kerampont – BP 80518
22305 Lannion Cedex, France
{bosc, hadjali, pivert}@enssat.fr

Abstract

In this paper we address the problem of query failure in the context of flexible querying. We propose a fuzzy set-based approach for relaxing queries involving gradual predicates. This approach relies on the notion of proximity relation which is defined in an absolute way. We show how such proximity relation allows for transforming a given predicate into an enlarged one. The resulting predicate is semantically not far from the original one and it is obtained by a simple fuzzy arithmetic operation. The main features of the weakening mechanism are investigated and a comparative study with some methods proposed for the purpose of fuzzy query weakening is presented as well. Last, an example is provided to illustrate our proposal in the case of conjunctive queries.

Keywords. Cooperative answering, flexible query, fuzzy query weakening, proximity relation, fuzzy interval.

1 Introduction

Since the early 90's, there is an increasing interest in building intelligent information systems endowed with some cooperative behavior [12]. The most well-known issue approached in this field is the "*empty answer problem*", that is, the problem of providing the user with some alternative data when there is no data fitting his/her query. Several approaches have been proposed to deal with this issue. The *relaxation paradigm* [3][11] is one of the basic cooperative techniques used in most of such approaches. In the Boolean querying framework, query relaxation consists in expanding the user query by

replacing some query conditions by more general conditions or by just eliminating some conditions. This allows the database to return answers related to the original query that are more convenient than an empty answer. Let us also mention that other approaches propose *knowledge discovery-based solutions* to this problem, see for instance [14][15][16].

On the other hand, relying on *fuzzy queries* has the main advantage of diminishing the risk of empty answers. Indeed, fuzzy queries are based on preferences and retrieve elements that are more or less satisfactory rather than ideal. However, it may still happen that the database does not contain any element that satisfies, even partially, the criteria formulated by the user. Then, an additional relaxation must be performed on the fuzzy query to avoid such empty answers. This can be accomplished by replacing fuzzy predicates involved in the query with weakened ones. The resulting query is then less restrictive and more tolerant.

In the fuzzy framework, query weakening consists in modifying the constraints contained in the query in order to obtain a less restrictive variant. Such a modification can be achieved by applying a basic transformation to all the predicates of the query or to some of them. Let us note that fuzzy query weakening has not received too much attention in the literature. Very few works are concerned with this problem. The study done by Andreasen and Pivert [1] is considered as a pioneering work in this area. Their approach is based on a transformation that uses a particular *linguistic modifier*. More recently, in [5], another solution has been proposed. It is based on a particular *tolerance relation* modeled by a *parametric relative proximity relation*. This notion of proximity relation, which originates from qualitative reasoning about fuzzy orders of magnitude [13], is intended for defining a set of predicates that are close, semantically speaking, to a given predicate P . This approach is significantly different from the previous one. It provides the *semantic basis* for defining a stopping criterion of the iterative weakening process. Let us also mention the work done in [17] where the authors consider queries addressed to data summaries and propose a method based on a specified distance to repair failing queries. Repairing query appears as relaxing the constraints of the retrieval since the resulting query, called substituting query, is more permissive than the initial one. See also the platform PRETI [8] which includes a flexible querying module that is endowed with an empirical method to avoid empty answers to user requests expressing a search for houses to let.

It is well known that there are two points of view which can be considered to compare numbers and thus orders of magnitude x and y on the real line [9]. We can evaluate the extent to which the difference $x - y$ is *large*, *small* or *close* to 0; this is the absolute comparative approach. Or, we may use relative orders of magnitude, i.e., evaluate to what extent the ratio x/y is *close* to 1 or not. In [9] and [13], it has been pointed out that those two kinds of proximity relation can be applied to define a fuzzy set of values that are close to some real-valued x . As emphasized above, we have shown in [5] that a particular *relative proximity relation* can constitute a tool to generate enlarged predicates that are close, semantically speaking, to a given predicate P . Thus, a relative proximity relation-based approach for fuzzy query weakening has been proposed. In this

paper, we consider a particular *absolute proximity relation* and show how such proximity relation is appropriate for the purpose of relaxing fuzzy queries. The main features of the weakening mechanism resulting from the use of this kind of proximity are investigated as well.

The paper is structured as follows. The next section recalls the problem of fuzzy query weakening on the one hand, and presents some methods that are proposed to solve it on the other hand. Section 3 shows how an absolute proximity relation can be used for generating more permissive fuzzy predicates and for achieving query relaxation. In section 4, we provide a comparative study of three techniques to relax fuzzy queries. In section 5, the case of conjunctive fuzzy queries is investigated and an illustrative example is provided. Last, we conclude and outline some future works.

2 Fuzzy Query Weakening

Weakening a "failing" fuzzy query consists in modifying the constraints involved in the query in order to obtain a less restrictive variant. Let Q be a fuzzy query of the form P_1 and P_2 and ... and P_k (where P_i is a fuzzy predicate), and assume that the set of answers to Q is empty. A natural way to relax Q , in order to obtain a non-empty set of answers, is to apply a basic uniform transformation to each predicate P_i . This transformation process can be accomplished iteratively if necessary. Some desirable properties are required for any transformation T when applied to a predicate P ($T(P)$ representing the modified predicate):

- C_1 : T does not decrease the membership degree for any element of the domain, i.e., $\forall u \in \text{domain}(A), \mu_{T(P)}(u) \geq \mu_P(u)$ where A denotes the attribute concerned by P ;
- C_2 : T extends the support $S(P)$ of the fuzzy predicate P , i.e. $S(P) = \{u \mid \mu_P(u) > 0\}$ $\hat{=} S(T(P)) = \{u \mid \mu_{T(P)}(u) > 0\}$;
- C_3 : T preserves the specificity of the fuzzy predicate P , i.e. $C(P) = \{u \mid \mu_P(u) = 1\} = C(T(P)) = \{u \mid \mu_{T(P)}(u) = 1\}$.

Then, if P is a fuzzy predicate represented by the trapezoidal membership function (*t.m.f.*) (A, B, a, b) , the desired transformation T is such that $P' = T(P) = (A, B, T(A, a), T(B, b))$, as described in Figure 1.

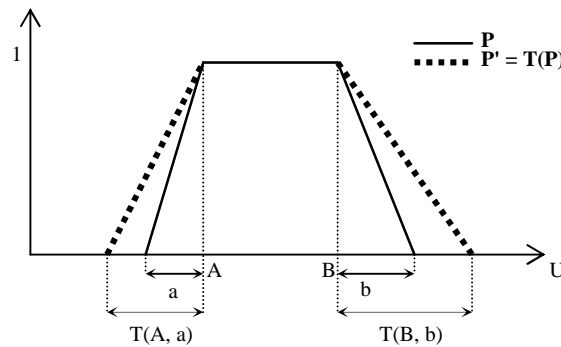


Figure 1. The basic transformation

As mentioned in the introduction, very few studies exist to deal with the issue of query weakening in a fuzzy setting. In each of them, a specific basic transformation is proposed. In the approach by Andreasen and Pivert [1], the transformation is based on a particular *linguistic modifier*. On the other hand, in the more recent one [5], the relaxation strategy makes use of a particular *proximity relation* expressed by a convenient *parametric fuzzy closeness relation*.

The rest of this section is devoted to the presentation of the two above approaches.

2.1 Linguistic Modifier-based Approach

To illustrate this approach, let us consider a query Q that involves a single fuzzy predicate, i.e. $Q = P$. As pointed out in [1], one way to weaken such a query is to apply a linguistic modifier to the fuzzy term P . Such modifier must have an expansive effect on the membership function associated to P . For instance, the query "find the employees who are *young*" can be transformed into "find the employees who are *more-or-less young*". In [7] Bouchon-Meunier has proposed a family of linguistic modifiers which have interesting properties. Especially, one modifier of this family, called *n-rather*, is of particular interest for the purpose of query weakening.

Definition 1. Let P be a fuzzy predicate represented by (A, B, a, b) . The linguistic modifier *n-rather* is defined such that

$$\mathbf{n}\text{-rather}(P) = (A, B, a/\mathbf{n}, b/\mathbf{n}),$$

with $\mathbf{n} \in [1/2, 1[$.

Now, according to Figure 1, the transformation T based on this modifier is such that $T(A, a) = a/\mathbf{n}$ and $T(B, b) = b/\mathbf{n}$. Denoting by $a' = a/\mathbf{n}$ and $b' = b/\mathbf{n}$, we can write $a' = a + \mathbf{q}a$ and $b' = b + \mathbf{q}b$, with $\mathbf{q} = (1 - \mathbf{n})/\mathbf{n}$ ($\mathbf{q} \in]0, 1[$). As can be seen, this transformation satisfies required properties C_1 to C_3 . Furthermore, the resulting weakening effects in the left and right sides (i.e., $\mathbf{q}a$ and $\mathbf{q}b$ respectively) are obtained on the basis of the same parameter \mathbf{q} . This is why the approach is said to be *quasi-symmetric* (it is *symmetric* if $a = b$ holds). Let us point out that this modifier is intended simply to perform a technical transformation, but does not have a clear inherent semantics.

Principle of the approach. Let us now explain how this modifier can be used to weaken a query Q . If the set of answers is empty, Q is transformed into $Q_1 = \text{rather}(P)$ and the process can be repeated n times until the answer to the question $Q_n = \text{rather}(\text{rather}(\dots \text{rather}(P)\dots))$ is not empty. In practice, the difficulty when applying this technique concerns its *semantic limits*. Namely, what is the maximum number of weakening steps that is acceptable according to the user, i.e., such that the final modified query Q_n is *not too far*, semantically speaking, from the original one. Indeed, no intrinsic criterion is provided for stopping the iterative process.

To overcome that problem, one solution consists in asking the user to specify, along with his/her query, a fuzzy set F_p of *more or less non-authorized* values in the related domain. Hence, the satisfaction degree of an element u becomes $\min(\mu_{Q_i}(u), 1 - \mu_{F_p}(u))$

with respect to the modified query Q_i resulting from i weakening steps. The weakening process will now stop when the answer is non-empty or when the core of the complementary of the support of Q_i is included in the core of F_P (i.e., $\min(\mu_{Q_i}(u), 1 - \mu_{F_P}(u)) = 0$).

2.2 Fuzzy Relative Closeness-based Approach

In the framework of a study about relative orders of magnitude in qualitative reasoning, a fuzzy set-based semantics has been proposed to closeness, negligibility and comparability relations [13]. It was shown that such relations can be represented by means of fuzzy relations controlled by tolerance parameters. The idea of relative closeness which expresses an approximate equality between two real numbers x and y , can be captured by the following definition.

Definition 2. The *closeness relation* (Cl) is a reflexive and symmetric fuzzy relation such that:

$$\mu_{Cl}(x, y) = \mu_M(x/y).$$

M is called a *tolerance parameter* and its characteristic function μ_M is that of a fuzzy number "close to 1", such that:

- i) $\mu_M(1) = 1$ (since x is close to x);
- ii) $\mu_M(t) = 0$ if $t \neq 0$ (assuming that two numbers which are close should have the same sign);
- iii) $\mu_M(t) = \mu_M(1/t)$ (since closeness is naturally symmetric, i.e., $\mu_{Cl}(x, y) = \mu_{Cl}(y, x)$).
This property implies that the support $S(M)$ is symmetric and is of the form $[1 - \epsilon, 1/(1 - \epsilon)]$.

Strict (or classical) equality is recovered for $M = I$ defined as $\mu_I(x/y) = 1$ if $x = y$ and $\mu_I(x/y) = 0$ otherwise. According to this point of view, we evaluate the extent to which the ratio x/y is close to 1. The closer x and y are, the closer to 1 x/y must be according to M . In what follows, $Cl[M]$ denotes the closeness relation parameterized by the tolerance indicator M .

Semantic properties of M. It has been demonstrated in [13] that the fuzzy number M which parameterizes closeness (and negligibility) should be chosen such that its support $S(M)$ lies in the *validity interval* $\forall = [(\bar{0}5 - 1)/2, (\bar{0}5 + 1)/2]$ in order to ensure that the closeness relation is more restrictive than the relation "not negligible". This means that if the support of a tolerance parameter associated with a closeness relation Cl is not included in \forall , then the relation Cl is not in agreement with the intuitive semantics underlying this notion. It is worth noticing that the *validity interval* \forall plays a key role in the query weakening process. As it will be shown later, it provides the basis for defining a stopping criterion of an iterative weakening process.

Principle of the approach. As pointed out in [5], a way to perform query weakening is to apply a tolerance relation to the fuzzy requirements involved in the query. A particular tolerance relation which is of interest in the context of query weakening can be conveniently modeled by the proposed parameterized closeness relation. Let us consider a query which only involves one fuzzy predicate P , and a closeness relation parameterized by a tolerance indicator M , $Cl[M]$. Now, to relax this query we replace the predicate P by an enlarged fuzzy predicate P' defined as follows:

$$\forall u \in U, \mu_{P'}(u) = \sup_{v \in U} \min(\mu_P(v), \mu_{Cl[M]}(u, v)).$$

Using the extension principle, it is easy to check that $P' = P \overset{\circ}{\bullet} M$, where $\overset{\circ}{\bullet}$ is the product operation extended to fuzzy numbers, see [10]. Clearly, the relative closeness-based transformation leads to a modified predicate P' which gathers the elements of P and the elements outside P which are somewhat close to an element in P . Hence, this approach conveys a clear inherent semantics.

In a formal way, the transformation T is such that

$$T(P) = P' = P \circ Cl[M] = P \overset{\circ}{\bullet} M,$$

where \circ stands for the fuzzy composition operation. Let $P = (A, B, a, b)$ and $M = (1, 1, \mathbf{e}, \mathbf{e}/(1 - \mathbf{e}))$ where \mathbf{e} stands for the *relative tolerance value* and lies in $[0, (3 - \sqrt{5})/2]$ (this interval results from the inclusion $S(M) \overset{\circ}{\bullet} \forall$, see [13]). The modified predicate P' is such that $P' = (A, B, a + A\mathbf{e}, b + B\mathbf{e}/(1 - \mathbf{e}))$ by applying the above arithmetic formula. We can easily check that the desirable properties C_1 to C_3 are satisfied by P' . Namely, we have: i) $\forall u, \mu_{P'}(u) \geq \mu_P(u)$; ii) $S(P) \overset{\circ}{\bullet} S(P')$; iii) $C(P) = C(P')$.

Now, according to Figure 1, the following equalities hold: $T(A, a) = a + A\mathbf{e}$ and $T(B, b) = b + B\mathbf{e}/(1 - \mathbf{e})$. The quantity $A\mathbf{e}$ (respectively $B\mathbf{e}/(1 - \mathbf{e})$) represents the relaxation intensity in the left (respectively right) part of the membership function of P . Since $B\mathbf{e}/(1 - \mathbf{e}) > A\mathbf{e}$ ($\forall \mathbf{e} \in]0, (3 - \sqrt{5})/2[$), then the relaxation is stronger in the right part than in the left part. This means that the weakening mechanism is of a *non-symmetrical* nature. Let us also emphasize that the maximal relaxation, denoted P^{RP-max} , of a predicate P can be reached for the tolerance value $\mathbf{e}_{max} = (3 - \sqrt{5})/2 \approx 0.38$. Hence, $P^{RP-max} = (A, B, a + A\mathbf{e}_{max}, b + B\mathbf{e}_{max}/(1 - \mathbf{e}_{max}))$.

In practice, if Q is a query containing a single predicate P (i.e., $Q = P$) and if the set of answers to Q is empty, then Q is relaxed by transforming it into $Q_1 = P \overset{\circ}{\bullet} M$. This transformation is repeated n times until the answer to the question $Q_n = P \overset{\circ}{\bullet} M^n$ is not empty. Now, in order to ensure that the query Q_n is semantically close enough to the original one, the support of M^n should be included in the validity interval \forall . Then, the above iterative procedure will stop either when the answer is non-empty, or when $S(M^n) \overset{\circ}{\bullet} \forall$, see Algorithm 1 (where S_{Q_i} stands for the set of answers to Q_i). As we can see, the main advantage of this approach is the fact that it provides *semantic limits* for controlling the relaxation process.

Algorithm 1.

```

let Q = P
let ε be a relative tolerance value (* e Î [0, (3 - √5)/2] *)
i := 0 (* i denotes the number of weakening steps *)
Qi := Q
compute ΣQi
while (ΣQi = ∅ and S(Mi+1) ⊆ V) do
  begin
    i := i+1
    Qi := P ⊗ Mi
    compute ΣQi
  end
if ΣQi ≠ ∅ then return ΣQi endif.

```

3 Absolute Proximity Relation-based Approach to Fuzzy Query Weakening

The purpose of this section is twofold. First, the notion of an *absolute proximity* relation is introduced. Then, the method based on this proximity relation to address the problem of query weakening is discussed.

3.1 Absolute Proximity Relation

Definition 3. An *absolute proximity relation* is an approximate equality relation which can be modeled by a fuzzy relation E of the form [9]:

$$\mu_E(x, y) = \mu_Z(x - y),$$

which only depends on the value of the difference $x - y$, and where Z , called a *tolerance parameter*, is a fuzzy interval centered in 0 , such that: i) $\mu_Z(r) = \mu_Z(-r)$; ii) $\mu_Z(0) = 1$; iii) its support $S(Z) = \{r \mid \mu_Z(r) > 0\}$ is bounded and is denoted by $[-d, d]$ where d is a real number.

Property (i) ensures the symmetry of the approximate equality relation ($\mu_E(x, y) = \mu_E(y, x)$); (ii) expresses that x is approximately equal to itself with the degree 1 . Here we evaluate to what extent the amount $x - y$ is close to 0 : the closer x is to y , the closer $x - y$ and 0 are. Classical equality is recovered for $Z = \mathbf{0}$ defined as $\mu_{\mathbf{0}}(x - y) = 1$ if $x = y$ and $\mu_{\mathbf{0}}(x - y) = 0$ otherwise. In terms of *t.m.f.*, the parameter Z can be expressed by $(0, 0, d, d)$.

Other interesting properties of the parameterized relation E are available in [9]. Furthermore, we shall write $E[Z]$ to denote the proximity relation E parameterized by Z .

Principle of the approach. The transformation explored in section 2.2 aims at finding a set of predicates that are close to a given predicate P . This is achieved by composing the predicate P with an appropriate relative proximity relation expressed by the fuzzy closeness relation $Cl[M]$. To do so, it is also possible to use an absolute proximity relation. Let us sketch how this can be done.

Assume that P is a fuzzy predicate and $E[Z]$ an absolute proximity relation. The predicate P can be relaxed into a fuzzy predicate P' defined in the following way:

$$\begin{aligned}\mu_{P'}(u) &= \sup_{v \in \mathcal{U}} \min(\mu_P(v), \mu_{E[Z]}(u, v)), \\ &= \sup_{v \in \mathcal{U}} \min(\mu_P(v), \mu_Z(u - v)), \\ &= \mu_{P \hat{\wedge} Z}(u), \text{ observing that } v + (u - v) = u.\end{aligned}$$

This means that $P' = P \hat{\wedge} Z$, where $\hat{\wedge}$ is the addition operation extended to fuzzy numbers [10]. As we can see, P' contains P and the elements outside P which are in the neighborhood of an element of P . Hence, the transformation is endowed with a clear semantics induced by the semantics underlying the relation $E[Z]$.

Formally, this transformation writes

$$T(P) = P' = P \circ E[Z] = P \hat{\wedge} Z$$

Let $P = (A, B, a, b)$ and $Z = (0, 0, \mathbf{d}, \mathbf{d})$ where \mathbf{d} stands for an *absolute tolerance value*. Using the above arithmetic formula, we obtain $P' = (A, B, a + \mathbf{d}, b + \mathbf{d})$. Figure 2 illustrates this transformation. It is easy to check that T is in agreement with the requirements C_1 to C_3 . The following equalities hold as well: $T(A, a) = a + \mathbf{d}$ and $T(B, b) = b + \mathbf{d}$. This means that the weakening effect in the left and right sides of the *t.m.f.* of P is the same and it is quantified by the scalar \mathbf{d} . Due to this equality, the resulting weakening is then of a *symmetrical* nature.

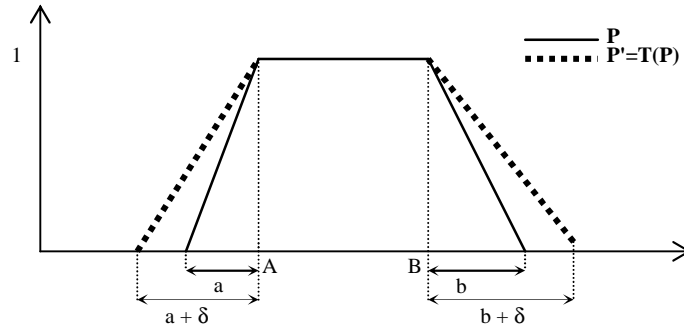


Figure 2. Absolute proximity relation-based weakening

Particular cases: Let us emphasize that for some kind of fuzzy predicates to be relaxed, the property of symmetry of the tolerance indicator Z is not required. Consider, for instance, the predicate $P = (0, 25, 0, 10)$ expressing the concept "young", as depicted in Figure 3.

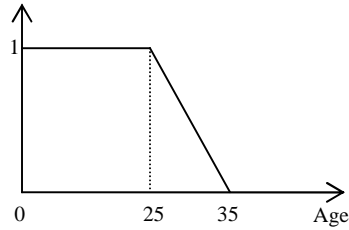


Figure 3. Fuzzy predicate "young".

Weakening P comes down to increase the cardinality of its support $S(P)$. This can be only done in the right side of the *t.m.f.* of P . Then, the appropriate family of the tolerance indicators will be of the form $Z = (0, 0, 0, \mathbf{d})$.

Let us now show how this kind of proximity relation can be used to relax a query Q containing one predicate P ($Q = P$). If the set of answers to Q is empty, then Q is transformed into $Q_1 = P \mathbf{\hat{A}} Z$. This progressive relaxation mechanism can be applied iteratively until the answer to the resulting query $Q_n = P \mathbf{\hat{A}} nZ$ is not empty. This strategy to single-predicate queries provides an implicit measure of *nearness* such that: Q_k is nearer to Q than Q_l if $k < l$. From a practical point of view, this mechanism is very simple to implement. However, no information is provided about the *semantic limits*. Indeed, no intrinsic criterion is attached to this transformation which would enable to stop the iterative process when the answer still remains empty.

Controlling the Relaxation. To enable some control over the relaxation process, once again we can use the fuzzy set F_p of more or less forbidden values in the related domain (mentioned in section 2.1). Then, the satisfaction degree of an element u becomes $\min(\mu_{Q_i}(u), 1 - \mu_{F_p}(u))$ with respect to the modified query Q_i resulting from i weakening steps. Thus the weakening process will now stop when the answer to Q_i is not empty ($S_{Q_i} \neq \mathbf{\emptyset}$) or when the *core* of the complementary of the support of Q_i is included in the *core* of F_p (i.e., $\min(\mu_{Q_i}(u), 1 - \mu_{F_p}(u)) = 0$).

This weakening technique can be sketched by Algorithm 2 (where S_{Q_i} stands for the set of answers to Q_i , $S(Q_i)^c$ for the complementary of the support of Q_i and $C(A)$ for the core of A , i.e., $\{u \mid \mu_A(u) = 1\}$).

Algorithm 2.

```

let Q := P
let  $\delta$  be an absolute tolerance value    (* Z = (0, 0,  $\delta$ ,  $\delta$ ) *)
i := 0
Qi := Q
compute  $\Sigma_{Q_i}$ 
while ( $\Sigma_{Q_i} = \emptyset$  and ( $C(S(Q_i)^c) \circ C(F_p)$ )) do
  begin
    i := i+1
    Qi := P  $\oplus$  i-Z
    compute  $\Sigma_{Q_i}$ 
  end
if  $\Sigma_{Q_i} \neq \emptyset$  then return  $\Sigma_{Q_i}$  endif.

```

4 A Comparative Study

In the following, we first investigate the main features of the absolute proximity relation-based approach. For the relative proximity relation (respectively linguistic modifier) based approach, and due to space limitation, we only discuss their features in a summarized way (a complete study is available in [5]). Then, we provide a comparative table with respect to some criteria that will be further given.

As it is illustrated in Figure 4, the *slopes* and the *relative position* of the membership function have no impact on the weakening effect when using the absolute proximity relation-based approach. However, the *attribute domain* is identified as a major factor affecting the weakening because d is an *absolute value* which is added and subtracted (d will be different for the attribute "age" and the attribute "salary").

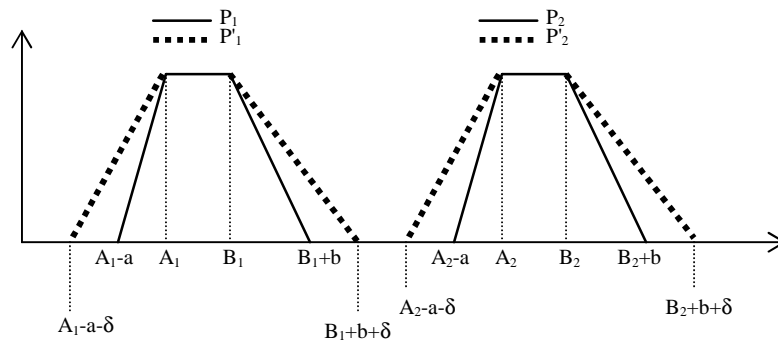


Figure 4. Impact of the slopes and the relative position of the membership functions
 $(a_1=a_2=a < b_1=b_2=b \Rightarrow a'_1=a'_2=a+\delta < b'_1=b'_2=b+\delta)$.

Now in practice, it is of great interest to compare the behaviors of the three weakening methods mentioned above in order to design some kind of "guide" enabling the user to choose which method is the most suitable. To do this, we have listed five criteria that seem of major importance from a user point of view:

- i) *Preservation/modification of the specificity of the attribute*: it consists to verify whether the set of typical values (i.e., the core) of the predicate is modified or not.
- ii) *Symmetric/non-symmetric weakening*: checking whether the weakening effect in the right and left parts of the *t.m.f.* of the predicate is similar or not.
- iii) *Semantic control of the relaxation*: it concerns the criteria that would allow for controlling and stopping the weakening process.
- iv) *Factors related to the domain and the predicate*: checking whether the attribute domain and the shape (or relative position) of the predicate membership function can have some impact on the weakening effect.
- v) *Applicability in the crisp case*: it consists in verifying if the transformation considered is still valid for predicates expressed as traditional intervals.

In Table 1, we summarize the behavior of each query weakening technique with respect to the above five criteria.

Table 1. Comparison

Criteria	Linguistic modifier-based approach	Relative closeness relation-based approach	Absolute proximity relation-based approach
(i)	Attribute specificity preserved	Attribute specificity preserved	Attribute specificity preserved
(ii)	Symmetrical weakening under certain conditions	Non symmetrical weakening	Symmetrical weakening by nature
(iii)	No intrinsic semantic limits	semantic limits provided	No intrinsic semantic limits
(iv)	Attribute domain-independent and predicate membership function-dependent	Attribute domain-independent and predicate membership function-dependent	Attribute domain-dependent and predicate membership function-independent
(v)	Inappropriate in the crisp case	Still effective in the crisp case	Still effective in the crisp case

As it is shown in Table 1, the two advantages of the absolute proximity relation-based approach with respect to the linguistic modifier-based one are the symmetrical nature of its weakening effect and its applicability when relaxing conventional queries. However, the most interesting features of the relative closeness-based approach remains the rigorous semantic limits for controlling the query relaxation level that it provides.

5 Some Issues Related to Conjunctive Fuzzy Queries Weakening

A conjunctive fuzzy query Q is of the form P_1 *and* P_2 *and* ... *and* P_k , where the conjunction is interpreted by the 'min' operator (we can use any other t -norm for interpreting this connector) and P_i is a fuzzy predicate. Note that the fuzzy set framework can provide two types of approaches to weaken a conjunctive fuzzy query: the *term modification-based* approach (that is the concern of this paper) and the *connector modification-based* approach. The latter is based on the replacement of one or more connectors by less restrictive variants along the scale with disjunction as the least and conjunction as the most restrictive connector. We will not consider this approach here. For more details, see [2].

In the case of the *term modification-based* approach, two strategies can be envisaged for the weakening procedure:

- i) a *global* query modification which consists in applying uniformly the basic transformation to all the predicates in the query. Given a transformation T and a conjunctive query $Q = P_1$ *and* P_2 *and* ... *and* P_k , the set of revised queries related to Q resulting from applying T is

$$\{T^i(P_1) \text{ and } T^i(P_2) \text{ and } \dots T^i(P_k)\},$$

where $i \geq 0$ and T^i means that T is applied i times. This strategy is simple but conflicts somewhat with our aim, that is, to find the closest revised query.

- ii) a *local* query modification which affects only some predicates (or sub-queries). Most of the time, only a part of the query is responsible for the empty answers. As a consequence, it is not necessary to modify all the predicates in the query to avoid this problem. In such cases, local strategy seems more suitable and results in modified queries that are closer to the original one than the modified ones provided by the global strategy. Another argument in a favor of the local strategy is its ability for explaining the cause of the initial empty answer (indeed, only the modified predicates involved in the final revised query are responsible for the initial empty answer).

In the following, we only focus on this latter strategy of relaxation.

5.1 Local query weakening

In this case, the basic transformation applies only to subqueries. Given a transformation T and a query $Q = P_1$ *and* P_2 *and* ... *and* P_k , the set of modifications of Q by T is

$$\{T^{i_1}(P_1) \text{ and } T^{i_2}(P_2) \text{ and } \dots T^{i_k}(P_k)\},$$

where $i_h \geq 0$ and T^{i_h} means that the transformation T is applied i_h times. Assume that all conditions involved in Q are of the same importance for the user, a total ordering (\prec)

between the revised queries related to Q can be defined on the basis of the number of the applications of the transformation T . Then, we have

$$Q' \prec Q'' \text{ if } \text{count}(T \text{ in } Q') < \text{count}(T \text{ in } Q'').$$

This ordering allows for introducing a *semantic distance* between queries.

The total ordering induced by the transformation defines a lattice of modified queries. For instance, the lattice associated with the weakening of the query " $P_1 \tilde{U} P_2 \tilde{U} P_3$ " (with the symbol \wedge stands for the operator 'and') is given by Figure 5.

In practice, three main issues must be dealt with when using this local strategy:

- i) Define a way to exploit the lattice of weakened queries.
- ii) Guarantee the property of equal relaxation for all fuzzy terms.
- iii) Study the user behavior with respect to the relaxation process, i.e., to what extent the user has to intervene in this process?

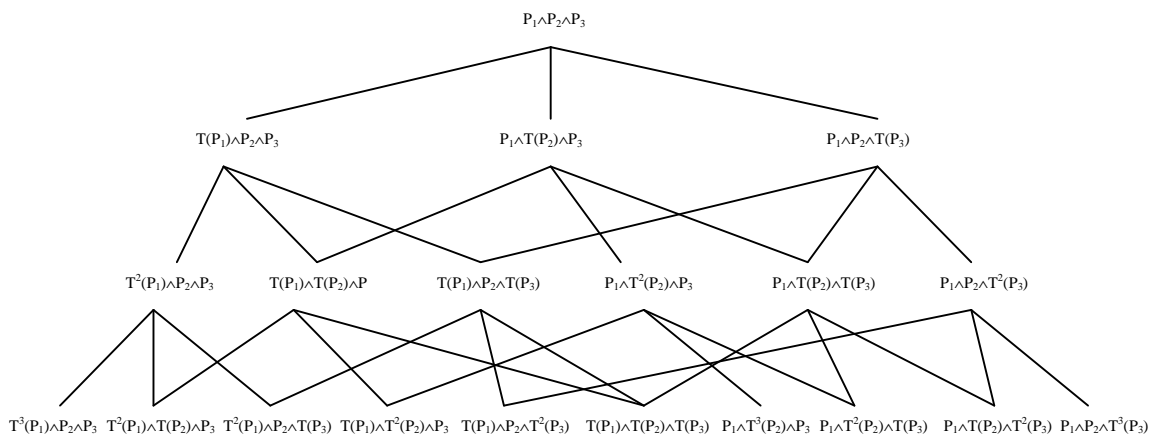


Figure 5. Lattice of relaxed queries (reduced to three levels)

5.1.1 Exploiting the lattice. We will not address this issue here, see [6] for more discussion. Let us, however, emphasize that scanning a lattice can be done either in breadth-first or in depth first. In our case, the depth-first way cannot fare well. The breadth-first way should fare well since it allows for finding a modified query with a non-empty answer that is as close as possible to initial query (according to the distance defined above).

To explore a lattice of relaxed queries, we propose the following way:

- i) First, ensure that each predicate has a non-empty support. Otherwise, relax all the predicates with empty supports.
- ii) If the answer to the query is still empty, we proceed from the left to the right by evaluating each weakened query belonging to the first level and we test the emptiness of its result. If all the answers are empty, we generate the weakened

queries of the second level and the same method is applied and so on. Otherwise, the first weakened query with a non-empty answer is returned.

5.1.2 Property of equal relaxation. As mentioned above, a semantic distance is induced by the ordering over the set of revised queries. In order that this distance makes sense, it is desirable that the transformation T fulfills the property of Equal Relaxation Effect (ERE) on all the fuzzy predicates. Several ways can be used for defining this property. A possible way is to consider the ratio of the lengths of the supports of the original and the modified predicates. This ratio must be of the same magnitude when a certain transformation T is applied. Let us denote $\Delta(P, T(P))$ this ratio when T is applied to P . We have

$$\mathbf{D}(P, T(P)) = \mathbf{U}(T(P))/\mathbf{U}(P),$$

where $\mathbf{U}(P)$ and $\mathbf{U}(T(P))$ represent the lengths of the supports of P and $T(P)$ respectively. A simple calculus enables to obtain (with $\mathbf{W} = \mathbf{B} - \mathbf{A} + \mathbf{b} + \mathbf{a}$):

$$\mathbf{D}(P, T(P)) = 1 + 2\mathbf{d}/\mathbf{W}.$$

Note that in the case where T is based on the fuzzy relative closeness (i.e., relative proximity relation), this ratio writes (with $\mathbf{h} = \mathbf{e}/(1 - \mathbf{e})$):

$$\mathbf{D}(P, T(P)) = 1 + (\mathbf{A}\mathbf{e} + \mathbf{B}\mathbf{h})/\mathbf{W}.$$

Now, given k predicates P_1, \dots, P_k , the equal weakening effect property for a set of transformations (T_1, \dots, T_k) can be expressed as follows:

$$\mathbf{D}(P_1, T_1(P_1)) = \mathbf{D}(P_2, T_2(P_2)) = \dots = \mathbf{D}(P_k, T_k(P_k)).$$

In what follows, we denote by AP-method (respectively RP-method) the absolute proximity relation (respectively relative proximity relation) based method for query weakening.

5.1.3 User behavior. It is very desirable that the user does not have to intervene in all the steps of the relaxation process of a retrieval information system. Indeed, a failing user query would lead to automatically run the cooperative answering strategies. Such strategies attempt to relax the original query and then to find alternative answers. In our relaxation process, the role of the user is only reduced to provide the maximal number, say n , of weakening steps that he/she authorizes. For the absolute tolerance values \mathbf{d}_j ($j=1,k$), they are automatically initialized following the method described below.

5.1.4. Practical computation of δ_j . To achieve this calculus, we first use the RP-method to determine which the predicate $P_i = (A_i, B_i, a_i, b_i)$ would reach the fastest its maximal relaxation (i.e., $P_i^{RP-max} = (A_i, B_i, a_i + A_i \cdot \epsilon_{max}, b_i + B_i \cdot \epsilon_{max}/(1 - \epsilon_{max}))$ see section 2.2). This can be done by assigning the quantity ϵ_{max}/n to the relative tolerance value of each P_j and checking whether the Global Relative Increment (denoted by GRI_j) over the weakening steps goes beyond the Maximal Relative Increment (denoted by MRI_j) for each predicate P_j ($j = 1,k$):

- The relative increment of the support of P_j at each weakening step is defined by $inc_j = (inc_{j_l} + inc_{j_r})/W_j$ with $W_j = A_j - B_j + a_j + b_j$, $inc_{j_l} = A_j \times e_j$ and $inc_{j_r} = B_j \times e_j / (1 - e_j)$. Hence, $GRI_j = n \times inc_j$.
- Now for each P_j , $MRI_j = (A_j \times e_{max} + B_j \times e_{max} / (1 - e_{max})) / W_j$.

In a second time, we consider the predicate P_i (resulting from the above step of calculus) and we estimate its maximal relaxation, denoted by P_i^{AP-max} , when using the AP-method. This estimation is possible by assuming that P_i^{AP-max} will not go beyond the maximal relaxation provided by the RP-method, i.e., $P_i^{AP-max} \subseteq P_i^{RP-max}$. Now, due to the symmetrical nature of AP-method, P_i^{AP-max} writes $(A_i, B_i, a_i + v_i, b_i + v_i)$ where v_i stands for the global relaxation over all the n steps and it is equal to $d_i \times n$. From the above inclusion, we deduce that $v_i = A_i \times e_{max}$ since $A_i \times e_{max} \leq B_i \times e_{max} / (1 - e_{max})$. Hence, d_i associated to P_i is equal to v_i/n . Last, by the ERE property we obtain the other tolerance values d_j for $j=1, k$ and $j \neq i$.

This computation process can be formalized in the two following steps:

Step1. *Applying the RP-method for searching the predicate P_i that reaches the fastest its maximal relaxation*

1. let $Q = P_1 \tilde{U} \dots \tilde{U} P_k$ be a fuzzy conjunctive query
2. choose a predicate $P_i = (A_i, B_i, a_i, b_i)$ and let $e_i = e_{max}/n$ (with $e_{max} @ 0.38$);
3. let $j := 1$;
 while $(j \neq k)$ and $(GRI_j \neq MRI_j)$ do
 begin
 $j := j + 1$
 end
4. if $j \neq k + 1$ then
 begin
 $i := j$, goto (2)
 end
5. return P_i .

Step2. *Computing the parameters d_j of the AP-method*

1. let $P_i = (A_i, B_i, a_i, b_i)$ the predicate provided by step1;
2. let $P_i^{AP-max} = (A_i, B_i, a_i + v_i, b_i + v_i)$ with $v_i = A_i \times e_{max}$;
3. compute the tolerance value d_i associated to P_i , i.e., $d_i = v_i/n$;
4. making use of the ERE property, we compute d_j for $j=1, k$ and $j \neq i$.

5.2 An Illustrative Example

To illustrate our proposal, we have tailored an example inspired from [4]. It concerns a big company that is organized in several departments and employs many persons. The relation of interest is described by three attributes that concern the *salary*, the *age* and

the *budget* of the department of an employee. The content of this relation is given in Table 2.

Table 2. Relation of the employees

Name	Salary (k€)	Age	Budget (k€)	$\mu_{P_1}(u)$	$\mu_{P_2}(v)$	$\mu_{P_3}(w)$
Dupont	2	48	38	1	0	0.8
Martin	1.7	46	34	0.4	0	1
Durant	1.3	45	32	0	0	0.6
Jones	1.2	37.5	24	0	0.5	0
Smith	1	34	20	0	0	0

Assume that a user wants to find the employees who satisfy the conditions: to be paid *around 2 k€* to be *about forty* and work in an *important* department. The gradual predicates *around_2*, *about_forty* and *important* are labels of fuzzy sets represented respectively by the following *t.m.f.*: $P_1 = (2, 2, 0.5, 0.5)$, $P_2 = (38, 42, 1, 1)$ and $P_3 = (34, 36, 10, 10)$ as depicted in Figure 6.

Then, the query of interest can simply write $Q = P_1 \hat{U} P_2 \hat{U} P_3$. As can be seen in Table 2, by composing the satisfaction degrees of each employee with respect to the three requirements expressed by P_1 , P_2 and P_3 , none of the employees satisfies the user query Q . Now, in order to return alternative answers to the user, we try to cooperate with him by relaxing his/her question. We first achieve this relaxation using the *AP-method*. Then, we consider the *RP-method*.

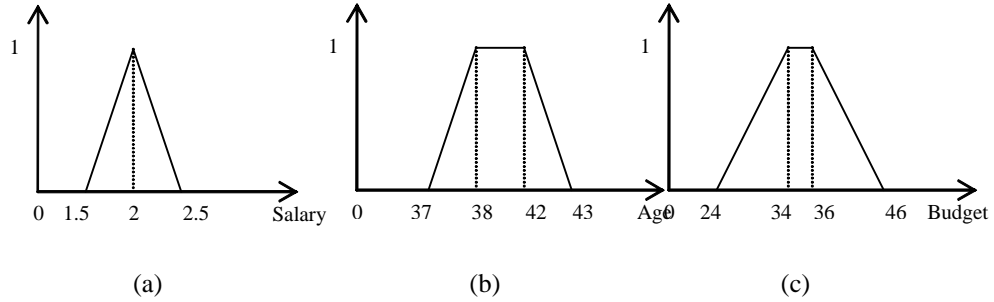


Figure 6. Fuzzy predicates: (a) *around_2*, (b) *about_forty*, (c) *important*.

5.2.1 Applying the AP-Method. Let us assume that the maximal number of weakening steps that the user authorizes is $n = 3$. Now, we proceed to the estimation of the absolute tolerance values d_j^i ($j = 1, 3$) associated to P_j as described in Section 5.1.

Estimation of δ_j . By step1, we show that the predicate P_1 is the one that will reach its maximal relaxation the fastest when using the RP-method. Then, the relative tolerance value $e_1 = e_{max}/n = 0.38/3 @ 0.12$.

Now by step2, we estimate the maximal relaxation $P_1^{AP-max} = (A_1, B_1, a_1 + \mathbf{v}_1, b_1 + \mathbf{v}_1) = (2, 2, 0.5 + \mathbf{v}_1, 0.5 + \mathbf{v}_1)$ with $\mathbf{v}_1 = A_1 \times \mathbf{e}_{max} = 2 \times 0.38 = 0.76$. This implies that $\delta_1 = 0.76/3 @ 0.25$. By the ERE property, we have $\mathbf{D}(P_1, T_1(P_1)) = \mathbf{D}(P_2, T_2(P_2)) = \mathbf{D}(P_3, T_3(P_3))$ which implies that $\mathbf{d}_1/W_1 = \mathbf{d}_2/W_2 = \mathbf{d}_3/W_3$. Hence, we obtain $\delta_2 = 1.5$ and $\delta_3 = 5.5$.

Relaxation Process. According to the method proposed in Section 5.1, we first transform the user query Q into the one of modified queries of the level 1 of the lattice (see Figure 5).

- Level 1: Q is transformed into $Q_1 = T(P_1)\hat{U}P_2\hat{U}P_3$ with $T(P_1) = (2, 2, 0.75, 0.75)$.

Table 3 summarizes the returned results when querying the database using Q_1 .

Unfortunately, the set of answers is still empty.

Table 3.

Name	Salary (k€)	Age	Budget (k€)	$\mu_{T(P_1)}(\mathbf{u})$	$\mu_{P_2}(\mathbf{v})$	$\mu_{P_3}(\mathbf{w})$	Satisfaction degree to Q_1
Dupont	2	48	38	1	0	0.8	0
Martin	1.7	46	34	0.6	0	1	0
Durant	1.3	45	32	0.067	0	0.6	0
Jones	1.2	37.5	24	0	0.5	0	0
Smith	1	34	20	0	0	0	0

We can easily show that all the remaining modified queries of this level (i.e., $P_1\hat{U}T(P_2)\hat{U}P_3$ and $P_1\hat{U}P_2\hat{U}T(P_3)$) also result in an empty answer. Then, we generate the weakened queries of the second level (see Figure 5).

- Level 2: all the modified queries in this level return an empty set of answers. Due to space limitation, we cannot go into the computation's details. However, we provide the results returned by the two modified queries $Q_2^1 = T(P_1)\hat{U}T(P_2)\hat{U}P_3$ and $Q_2^2 = P_1\hat{U}T^2(P_2)\hat{U}P_3$ with $T(P_2) = (38, 42, 2.5, 2.5)$ and $T^2(P_2) = (38, 42, 4, 4)$. In Table 4 (respectively Table 5), we report the results of Q_2^1 (respectively Q_2^2).

Table 4.

Name	Salary (k€)	Age	Budget (k€)	$\mu_{T(P_1)}(\mathbf{u})$	$\mu_{T(P_2)}(\mathbf{v})$	$\mu_{P_3}(\mathbf{w})$	Satisfaction degree to Q_2^1
Dupont	2	48	38	1	0	0.8	0
Martin	1.7	46	34	0.6	0	1	0
Durant	1.3	45	32	0.067	0	0.6	0
Jones	1.2	37.5	24	0	0.8	0	0
Smith	1	34	20	0	0	0	0

Table 5.

Name	Salary (k€)	Age	Budget (k€)	$\mu_{P_1}(u)$	$\mu_{T^2(P_2)}(v)$	$\mu_{P_3}(w)$	Satisfaction degree to Q_2^2
Dupont	2	48	38	1	0	0.8	0
Martin	1.7	46	34	0.4	0	1	0
Durant	1.3	45	32	0	0.25	0.6	0
Jones	1.2	37.5	24	0	0.875	0	0
Smith	1	34	20	0	0	0	0

An additional weakening step is then necessary. Hence, we consider the modified queries of the level 3 (see Figure 5).

- Level 3: let us first emphasize that the revised query $T^3(P_1)\hat{U}P_2 \hat{U}P_3$ always return an empty set of answers since no employee in the database (Table 2) satisfies the sub-query $P_2 \hat{U}P_3$. The first modified query that provides a non-empty answer is $Q_3 = T(P_1)\hat{U}T^2(P_2)\hat{U}P_3$, when scanning the lattice from the left to the right. Table 6 gives the satisfaction degrees to Q_3 of all the items contained in the database.

Table 6. Final results based on the AP-method

Name	Salary (k€)	Age	Budget (k€)	$\mu_{T(P_1)}(u)$	$\mu_{T^2(P_2)}(v)$	$\mu_{P_3}(w)$	Satisfaction degree to Q_3
Dupont	2	48	38	1	0	0.8	0
Martin	1.7	46	34	0.6	0	1	0
Durant	1.3	45	32	0.067	0.25	0.6	0.067
Jones	1.2	37.5	24	0	0.875	0	0
Smith	1	34	20	0	0	0	0

As can be seen, the employee *Durant* somewhat fits the requirements formulated in Q_3 , then the relaxation process stops and *Durant* is returned as an answer to the user (with the satisfaction degree 0.067).

5.2.2 Applying the RP-Method. As in the case of the AP-method, we start by estimating the relative tolerance values e_j ($j=2,3$) associated to P_j knowing that $e_1 @ 0.12$.

Estimation of ϵ_j . By the ERE property, we have $D(P_1, T_1(P_1)) = D(P_2, T_2(P_2)) = D(P_3, T_3(P_3))$ which implies that $(A_1 \times e_1 + B_1 \times h_1)/W_1 = (A_2 \times e_2 + B_2 \times h_2)/W_2 = (A_3 \times e_3 + B_3 \times h_3)/W_3$ (with $h_i = e_i/(1 - e_i)$ for $i=1,3$). Hence, we obtain $e_2 = 0.036$ and $e_3 = 0.14$.

Relaxation Process. We proceed in a similar way as in the case of the AP-Method. Namely, we generate the modified queries of each level of the lattice (see Figure 5) until one of them returns a non-empty set of answers.

- Level 1: Q can be transformed into $T(P_1)\hat{U}P_2 \hat{U}P_3$, $P_1\hat{U}T(P_2)\hat{U}P_3$ or $P_1\hat{U}P_2 \hat{U}T(P_3)$. All these relaxed variant return empty answers. Let us, for instance, consider the modified query $Q_1 = T(P_1)\hat{U}P_2 \hat{U}P_3$ with $T(P_1) = (2, 2, 0.74, 0.76)$.

In Table 7, we give the results returned by Q_1 with respect to the content of the database of Table 2. Unfortunately, no employee somewhat satisfies this query.

Table 7.

Name	Salary (k€)	Age	Budget (k€)	$\mu_{T(P_1)}(u)$	$\mu_{P_2}(v)$	$\mu_{P_3}(w)$	Satisfaction degree to Q_1
Dupont	2	48	38	1	0	0.8	0
Martin	1.7	46	34	0.59	0	1	0
Durant	1.3	45	32	0.05	0	0.6	0
Jones	1.2	37.5	24	0	0.5	0	0
Smith	1	34	20	0	0	0	0

- Level 2: We will not give all the details about the processing of the modified queries of this level. However, let us pinpoint that the modified query Q_2 expressed by $P_1 \tilde{U}T^2(P_2) \tilde{U}P_3$ is the first that provides a non-empty set of answers, when scanning the lattice from the left to the right. Below in Table 8, we summarize the result returned by Q_2 with $T^2(P_2) = (38, 42, 3.72, 4.10)$.

Table 8. Final results based on the RP-Method

Name	Salary (k€)	Age	Budget (k€)	$\mu_{P_1}(u)$	$\mu_{T^2(P_2)}(v)$	$\mu_{P_3}(w)$	Satisfaction degree to Q_2
Dupont	2	48	38	1	0	0.8	0
Martin	1.7	46	34	0.4	0.024	1	0.024
Durant	1.3	45	32	0	0.26	0.6	0
Jones	1.2	37.5	24	0	0.86	0	0
Smith	1	34	20	0	0	0	0

Thus, the weakening process ends successfully and returns the set of answers to the user, i.e., the employee *Martin* with the satisfaction degree 0.024 .

As can be seen, the RP-method leads to the desired results in two weakening steps, while the AP-method necessitates three steps. The reason is that, for the predicate P_2 and in step 2, the relaxation intensity in the right provided by the former method (i.e., 3.10) is greater than the relaxation intensity based on the latter method (i.e., 3).

6 Conclusion

An alternative fuzzy set-based approach for handling query failure is proposed. It contributes to enrich cooperative answering techniques in the context of usual database fuzzy querying. The proposed method is based on the notion of absolute proximity relation to define a predicate transformation. This transformation aims at finding a set of

the closest predicates, in the sense of the considered proximity relation, to a given predicate. The interesting feature of the approach is the fact that it operates only on the conditions involved in the initial user query without eliminating any condition or performing any summarizing operation on the database. This means that no information about the data in the database is required for relaxing queries. One direction of future works concern the implementation step and the test of the efficiency of the approach on some large practical examples.

References

- [1] T. Andreasen, O. Pivert, "On the weakening of fuzzy relational queries", Proc. 8th *Int. Symp. on Meth. for Intell. Syst.*, Charlotte, USA, 1994, pp. 144-151.
- [2] T. Andreasen, O. Pivert, "Improving answers to failing fuzzy relational queries," Proc. of the 6th *Int. Fuzzy Syst. Assoc. World Congress, IFSA*, São Paulo, Brazil, 1995, pp. 345-348.
- [3] F. Benamara, P. Saint Dizier, "Advanced relaxation for cooperative question answering", in: *New Directions in Question Answering*, Chap. 21, M. Maybury Ed., AAAI/MIT Press, 2004.
- [4] P. Bosc, B.B. Buckles, F.E. Petry, O. Pivert, "Fuzzy Databases," in: *Fuzzy Sets in Approximate Reasoning and Information Systems* (J.C. Bezdek, D. Dubois and H. Prade, Eds.), Kluwer Academic Publ., Dordrecht, Netherlands, 1999, pp. 403-468.
- [5] P. Bosc, A. HadjAli, O. Pivert, "Fuzzy closeness relation as a basis for weakening fuzzy relational queries," Proc. of 6th *Inter. Conf. FQAS*, Lyon, LNCS 3055, 2004, pp. 41-53.
- [6] P. Bosc, A. HadjAli, O. Pivert, "Towards a tolerance-based technique for cooperative answering of fuzzy queries against regular databases", Proc. of the 7th *Inter. Conf. CoopIS*, Cyprus, LNCS 3760, 2005, pp. 256-273.
- [7] B. Bouchon-Meunier, "Stability of linguistic modifiers compatible with a fuzzy logic", in: *Uncertainty in Intelligent Systems*, LNCS 313, 1988, pp. 63-70.
- [8] M. de Calmès, D. Dubois, E. Hullermeier, H. Prade, F. Sedes, "Flexibility and fuzzy case-based evaluation in querying: An illustration in an experimental setting", *Int. J. of Uncertainty, Fuzziness and Knowledge-based Systems*, 11(1), 2003, pp. 43-66.
- [9] D. Dubois, A. HadjAli, H. Prade, "Fuzzy qualitative reasoning with words", in: *Computing with Words* (P.P. Wang, Ed.), Volume 3 in a Series of Books on Intelligent Systems, John Wiley & Son, 2001, pp. 347-366.
- [10] D. Dubois, H. Prade, "Fundamentals of Fuzzy Sets", The Handbooks of Fuzzy Sets Series (

- D. Dubois, H. Prade, Eds), Vol. 3, Kluwer Academic Publ., Dordrecht, Netherlands, 2000.
- [11] T. Gaasterland, "Cooperative answering through controlled query relaxation", *IEEE Expert*, 12(5), Sep/Oct 1997, pp. 48-59.
- [12] T. Gaasterland, P. Godfrey, J. Minker, "An overview of cooperative answering", *Journal of Intelligent Information Systems*, 1(2), 1992, pp. 123-157.
- [13] A. HadjAli, D. Dubois, H. Prade, "Qualitative reasoning based on fuzzy relative orders of magnitude", *IEEE Transactions on Fuzzy Systems*, Vol. 11(1), 2003, pp. 9-23.
- [14] I. Muslea, "Machine learning for online query relaxation", Proc. of the 10th *Inter. Conf. of Knowledge and Discovery and Data mining*, KDD'2004, Washington, pp. 246-255.
- [15] J. Ozawa, K. Yamada, "Discovery of global knowledge in database for cooperative answering", Proc. of the 5th *IEEE Inter. Conf. on Fuzzy Systems*, 1995, pp. 849-852.
- [16] Z. W., Ras, D. Dardzinska, "Failing queries in distributed autonomous information systems", Proc. of *ISMIS*, LNAI No. 3488, 2005, pp. 152-160.
- [17] W.A. Voglozin, G. Raschia, L. Ughetto, N. Mouaddib, "Querying the SaintEtiq summaries: Dealing with null answers", Proc. of 14th *IEEE Inter. Conf. on Fuzzy Systems*, Reno (Nevada), USA, May 22-25, 2005, pp. 585-590.