Finding Similar Objects in Relational Databases — An Association-Based Fuzzy Approach

Olivier Pivert[1], Grégory Smits[2], and Hélène Jaudoin[1]

[1] Irisa – Enssat, University of Rennes 1
Technopole Anticipa 22305 Lannion Cedex France
[2] Irisa – Enssat, IUT Lannion
Technopole Anticipa 22305 Lannion Cedex France
{pivert,jaudoin}@enssat.fr, gregory.smits@univ-rennes1.fr

Abstract. This paper deals with the issue of extending the scope of a user query in order to retrieve objects which are similar to its "strict answers". The approach proposed exploits associations between database items, corresponding, e.g., to the presence of foreign keys in the database schema. Fuzzy concepts such as typicality, similarity and linguistic quantifiers are at the heart of the approach and make it possible to obtain a ranked list of similar answers.

1 Introduction

The practical need for endowing information systems with the ability to exhibit cooperative behavior has been recognized since the early '90s. As pointed out in [9], the main intent of cooperative systems is to provide correct, non-misleading and useful answers, rather than literal answers to user queries. Cooperative answers also aim at better serving the user's needs and expectations. The idea developed in this paper, inspired notably by Stefanidis *et al.* [13], consists in providing the user with answers which are not only "strict answers" to his/her query, but also objects that he/she might like ("You May Also Like") — as in recommender systems. In this paper, one investigates a fuzzy-set-based approach, which can also be seen as an extension of nearest neighbor queries where the notion of neighborhood considered is based on associations between entities (modeled for instance by foreign keys in a relational database context). As an introductory example, let us consider the bibliographic database composed of the relations:

– (A) $Author(a, name)$ of key a;
– (P) $Publi(t, title, journal)$ of key t;
– (K) $Keyword(w, word)$ of key w;
– (W) $Written\_by(t, a)$ of key (t, a), with foreign keys t and a;
– (D) $Deals\_with(t, w)$ of key (t, w), with foreign keys t and w.

Let us consider a query retrieving names of authors and let us assume that the user, after scanning the result, is interested in finding authors similar to one of the answers, say Codd. A possible meaning of "similar" in this context may be that the authors to be retrieved must publish in a set of journals similar to the set of journals where Codd publishes, must publish about a similar set of topics, and have a set of co-authors that is similar to Codd's. The approach we propose is based on a fuzzy comparison between the set of typical objects (journals, topics, co-authors) associated with a given target object (Codd in this example) and the set of typical objects associated with every other researcher present in the database. By doing so, a degree of matching can be measured for every researcher, which makes it possible to produce a top-k list of authors somewhat similar to Codd. Let us assume for instance that the fuzzy set of typical journals associated with Codd is $T_{Codd} = \{0.8/\text{PVLDB}, 0.6/\text{TKDE}, 0.3/\text{DKE}\}$. Then, taking into account the similarity based on journals, a researcher X will be considered all the more similar to Codd as his own set of typical journals T_X is close to T_{Codd} in the sense of an appropriate matching measure.

The remainder of the paper is structured as follows. Section 2 presents diverse approaches that may be used to compute the typical values of a multiset, i.e., that make it possible to convert a multiset E into a fuzzy set T describing the values that are the most typical in E. Section 3 discusses a sample of measures aimed at assessing the extent to which two fuzzy sets (of typical values, here) are similar. Such measures may be used to interpret the matching operator mentioned above. Section 4 discusses implementation aspects. Section 5 describes a preliminary experimentation that was carried out on the IMDb movie database. Section 6 discusses related work and situates our approach with respect to other proposals. Finally, Section 7 recalls the main contributions of the paper and outlines a few perspectives for future work.

2 Computing the Typical Values of a Multiset

Let us denote by f_i the relative frequency of a value x_i in a multiset E:

$$f_i = \frac{n_i}{n} \tag{1}$$

where n_i is the number of copies of x_i in E and n is the cardinality of E.

In order to assess the extent to which x_i is a typical value in E, two cases have to be taken into account: that where a metric — on which a similarity relation can be based — over the considered domain is available, and that where such a metric is not available and strict equality must be used. In any case, starting from a multiset E, the objective is to obtain a fuzzy set T such that $\forall x_i$, $\mu_T(x_i)$ expresses the extent to which x_i is typical in E.

2.1 Typicality Based on Strict Equality

In the absence of any similarity measure, an obvious solution is to take

$$\mu_T(x_i) = f_i. \tag{2}$$

Let us notice however that with this frequency-based approach, every element is considered somewhat typical. Those which have a low frequency get a low degree of typicality, but the elements which have a rather high frequency may also get a typicality degree significantly smaller than 1, since there are often several representative elements in a collection. Let us consider for instance a collection (multiset) of hundred animals including thirty dogs, thirty cats, and various other animals with only one occurrence each. The element "dog" has the frequency value 0.3, as well as the element "cat". Now, it could appear desirable to express that "dog" and "cat" are the two typical elements of the collection, to a high degree. One may then use:

$$\mu_T(x_i) = \mu_{most}(f_i) \tag{3}$$

where *most* is a fuzzy quantifier [17] whose general form is given in Figure 1. In order to get the desired behavior, one may use low values for δ and γ, for instance $\delta = 0.1$ and $\gamma = 0.5$ (which corresponds of course to a rather lax vision of *most*).

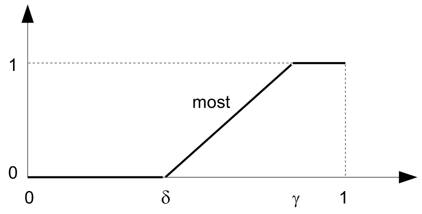

Fig. 1. A representation of *most*

2.2 Typicality Based on Similarity

When a similarity relation S over the considered domain is available, one may use the definition proposed by Dubois and Prade [8], which says, following Zadeh's interpretation [18], that an element x_i is all the more typical in a multiset E as it is both frequent in E and similar to most of the values of E:

$$\mu_T(x_i) = \frac{1}{n}\sum_{j=1}^{n} \mu_S(x_i,\, x_j) \tag{4}$$

with

$$\mu_S(x_i,\, x_j) = \max(0,\, \min(1,\, \frac{\alpha + \beta - d_{ij}}{\beta})), \tag{5}$$

where d_{ij} denotes the distance between x_i and x_j with respect to S, and the values α and β with $\alpha \leq \beta$ are positive real numbers which define a threshold of "indistinguishability" around each value x.

Example 1. Let us consider the multi-set (of cardinality $n = 30$):

$$E = \langle 1/0,\ 1/3,\ 1/4,\ 4/5,\ 7/6,\ 5/7,\ 3/8,\ 5/9,\ 2/12,\ 1/23 \rangle$$

where k/x_i means that element x_i has k copies in E. With $\alpha = 2$ and $\beta = 2$, and $d_{ij} = |x_i - x_j|$, one gets the fuzzy set of typical values:

$$T = \{0.05/0,\ 0.33/3,\ 0.52/4,\ 0.65/5,\ 0.77/6,\ 0.82/7,\ 0.73/8,\ 0.58/9,\ 0.15/12,\ 0.03/23\}$$

where μ/x_i means that element x_i belongs to T (i.e., is typical in E) to the degree μ.$\diamond$

Again, Formula (4) can be softened by applying a linguistic (fuzzy) quantifier *most* (meaning here "a significant proportion"):

$$\mu_T(x_i) = \mu_{most}\left(\frac{1}{n}\sum_{j=1}^{n}\mu_S(x_i,\ x_j)\right). \tag{6}$$

Here, the fuzzy quantifier *most* can be more drastic (for instance $\delta = 0.4$ and $\gamma = 0.8$) that in the strict equality case, since taking similarity into account generally leads to higher typicality degrees.

Example 2. Let us come back to the data of Example 1. Using the quantifier *most* defined by $\delta = 0.4$ and $\gamma = 0.8$, one gets:

$$T = \{0.3/4,\ 0.62/5,\ 0.92/6,\ 1/7,\ 0.82/8,\ 0.45/9\}.$$

Remark 1. In the case of nonnumerical attributes, defining the similarity measure (function μ_S) is not an easy task. A solution can be to use a domain ontology when it is available. See e.g. [2] where diverse similarity measures based on ontologies are discussed.

3 Fuzzy Matching Operator

Several interpretations of the condition E_1 *matches* E_2 — where E_1 and E_2 are two regular multisets of attribute values associated respectively with the target object and a candidate answer — can be thought of. The problem comes down to assessing the equality of two fuzzy sets, and many measures have been proposed for doing so, see, e.g., [12,5]. One may for instance:

– test the equality of the two fuzzy sets T_1 and T_2 of (more or less) typical elements in E_1 and E_2 respectively, for example by means of the Jaccard indice:

$$\mu_{matches}(E_1,\ E_2) = \frac{\sum_{x\in U}\min(\mu_{T_1}(x),\ \mu_{T_2}(x))}{\sum_{x\in U}\max(\mu_{T_1}(x),\ \mu_{T_2}(x))} \tag{7}$$

where U denotes the underlying domain of E_1 and E_2 — but this is rather drastic —, or by means of a measure such as:

$$\mu_{matches}(E_1,\ E_2) = \inf_{x\in U} 1 - |\mu_{T_1}(x) - \mu_{T_2}(x)|. \tag{8}$$

– check whether there exists at least one element which is typical both in E_1 and in E_2 (which corresponds to a rather lax view):

$$\mu_{matches}(E_1,\, E_2) = \sup_{x \in U} \min(\mu_{T_1}(x),\, \mu_{T_2}(x)). \tag{9}$$

– assess the extent to which most of the elements which are typical in E_1 are also typical in E_2 and reciprocally:

$$\mu_{matches}(E_1,\, E_2) = \min(\mu_{most \,\in\, T_2}(T_1),\, \mu_{most \,\in\, T_1}(T_2)). \tag{10}$$

The evaluation of Formula (10) is based on (one of) the interpretation(s) of fuzzy quantified statements of the form $Q\ X\ A\ are\ B$ where A and B are fuzzy predicates and Q is a fuzzy quantifier. See [17,15,16]. The most simple interpretation was proposed by Zadeh [17] and is based on the ratio of elements which are A and B among those which are A:

$$\mu(Q\ X\ A\ are\ B) = \mu_Q\left(\frac{\sum_{x \in X} \top(\mu_A(x),\, \mu_B(x))}{\sum_{x \in X} \mu_A(x)}\right) \tag{11}$$

where $\top$ denotes a triangular norm, for instance the minimum. Then, Equation (10) rewrites (taking $\top = \min$):

$$\begin{aligned}\mu_{matches}(E_1,\, E_2) = \min(\mu_{most}&\left(\frac{\sum_{x \in X} \min(\mu_{T_1}(x),\, \mu_{T_2}(x))}{\sum_{x \in X} \mu_{T_2}(x)}\right), \\ \mu_{most}&\left(\frac{\sum_{x \in X} \min(\mu_{T_1}(x),\, \mu_{T_2}(x))}{\sum_{x \in X} \mu_{T_1}(x)}\right)).\end{aligned} \tag{12}$$

Example 3. Let us consider the bibliographic database introduced in Section 1 and assume that two authors are considered similar if the typical sets of journals in which they publish are similar. Let us consider the typical sets:

$$T_1 = \{0.2/\text{PVLDB},\ 0.3/\text{TKDE},\ 0.6/\text{JIIS},\ 0.6/\text{DKE}\}$$

and

$$T_2 = \{0.3/\text{DKE},\ 0.4/\text{TODS},\ 0.6/\text{PVLDB},\ 0.8/\text{IJIS}\}.$$

The similarity degrees obtained using the previous measures are:

– with Formula (7): $\mu_{matches}(E_1,\, E_2) = \frac{0.5}{3.3} \approx 0.15$
– with Formula (8): $\mu_{matches}(E_1,\, E_2) = inf(0.6,\, 0.7,\, 0.4,\, 0.7,\, 0.6,\, 0.2) = 0.2$
– with Formula (9): $\mu_{matches}(E_1,\, E_2) = sup(0.2,\, 0,\, 0,\, 0.3,\, 0,\, 0) = 0.3$
– with Formula (12) using a quantifier "most" defined by $\delta = 0.1$ and $\gamma = 0.5$:
$$\begin{aligned}\mu_{matches}(E_1,\, E_2) &= min(\mu_{most}(\tfrac{0.5}{2.1}),\, \mu_{most}(\tfrac{0.5}{1.7})) \\ &= min(\mu_{most}(0.24),\, \mu_{most}(0.29)) \\ &= min(0.34,\, 0.48) = 0.48. \diamond\end{aligned}$$

Semantic proximity between values (if available) can also be taken into account during the computation of the similarity of two fuzzy sets. Such a matching measure, called *interchangeability*, is proposed in [4].

Remark 2. In the case where the query aimed at retrieving similar objects involves a conjunction of similarity conditions, the different degrees of matching may be aggregated by means of a triangular norm (e.g., the minimum operator), according to fuzzy set theory. However, some associations may be considered more important than others, and an aggregation operator such as the weighted average or the weighted minimum [7] can then be used instead of the minimum.

Remark 3. When the initial user query retrieves *several* objects (targets), one must look for similar items for each of these targets. Then, it seems reasonable to combine the different sets of similar items in a disjunctive manner: an object is in the extended result if it is similar to one strict answer (one target) at least. The degrees coming from the different sets of similar items are then combined by means of a triangular co-norm (e.g., the maximum operator).

4 Implementation Aspects

Let us first show how the different tools described in Sections 2 and 3 can be applied to the computation of a extended set of answers. First, let us emphasize the difficulty of defining a fully automated process in this case. Indeed, it does not seem possible, in general, to guess in what sense the end-user considers that an object is similar to another object (in the introductory example, for instance, is it because the authors have written papers on the same topics, and/or in the same journals, etc). The system needs some hints in order to derive the appropriate query. The outline of an interactive strategy could be as follows. After the strict answers are returned, the system asks the user "do you wish to get similar objects?". If his/her answer is "yes", the user is asked to check some boxes (predefined by a domain expert on the basis of primary key/foreign key constraints, in particular, or the discovery of so-called metapaths as defined in [14]) in order to specify his vision of "similar". For instance, in the context of the introductory example, the options could be: i) publications in similar journals, ii) publications on similar topics, iii) publications with similar co-authors.

Algorithm 1 describes the basic strategy for retrieving the objects similar to a target object c. In this algorithm, the $E_i(x)$'s are the multisets obtained by processing the n subqueries referring to x, and $T_i(x)$ is the fuzzy set of typical elements in $E_i(x)$.

Let m denote the cardinality of the relation containing the target objet c (and the objects one wants to retrieve). The previous algorithm implies to process $n \times m$ subqueries. An optimization consists in prefiltering the relation by means of a selection condition based on the typical sets associated with c, in order to avoid an exhaustive scan of the relation.

Example 4. Let us consider again the query from Section 1. Let us assume that the subquery:

select *journal* **from** P, W, A
where $P.t = W.t$ **and** $P.a = A.a$ **and** $name$ = 'Codd'

returns a multiset whose associated fuzzy set of typical elements is

$$T_1 = \{0.1/\text{JODS}, 0.2/\text{PVLDB}, 0.3/\text{TKDE}, 0.6/\text{JIIS}, 0.6/\text{DKE}\}.$$

Then, only the authors a belonging to the result of the following query should be considered:

select a **from** P, W **where** $P.a = W.a$
and *journal* **in** ('JODS', 'PVLDB', 'TKDE', 'JIIS', 'DKE')

since the other authors have no chance to be somewhat satisfactory (they do not share any journal with Codd).$\diamond$

```
Input: a target object $c$; $n$ specifications of multisets (i.e., $n$ subqueries);
a threshold $\alpha \in (0, 1]$
Output: a fuzzy set $S(c)$ of objects similar to $c$
begin
    $S(c) \leftarrow \emptyset$;
    for $i \leftarrow 1$ to $n$ do
        compute $E_i(c)$; compute $T_i(c)$ from $E_i(c)$;
    end
    foreach item $x$ in the relation concerned do
        for $i \leftarrow 1$ to $n$ do
            compute $E_i(x)$; compute $T_i(x)$ from $E_i(x)$;
            compute the degree of matching $\mu_i$ between $T_i(c)$ and $T_i$(x);
        end
        $\mu \leftarrow \min_{i=1..n}\ \mu_i$;
        if $\mu \geq \alpha$ then
            $S(c) \leftarrow S(c) \cup \{\mu/x\}$
        end
    end
end
```

Algorithm 1: Base algorithm

In the previous example, the selection condition is based on the support of the fuzzy set T_1. Notice that if one uses Equation (2) to compute the typicality degree, the support of a given $T_i(c)$ may be rather large. On the other hand, with Equation (3), one applies the measure of matching (cf. Section 3) to sets of objects that are *sufficiently typical* (i.e., whose frequency is sufficiently high). This makes it possible to use a more restrictive selection condition for filtering the relation. For instance, in the previous example, using a quantifier *most* defined by $\delta = \gamma = 0.25$, one would get the more selective prefiltering query:

select a **from** P, W **where** $P.a = W.a$
and *journal* **in** ('TKDE', 'JIIS', 'DKE')

since both JODS and PVLDB are not sufficiently typical with respect to the target objet to be taken into account while searching for similar authors.

5 Preliminary Experimentation

In order to check the efficiency and effectiveness of the approach, we performed a preliminary experimentation using the IMDb[1] movie database illustrated in Figure 2, where the cardinality of each table is given in brackets. In this context, queries may involve conditions on attributes such as actor's name, movie title, production date, etc.

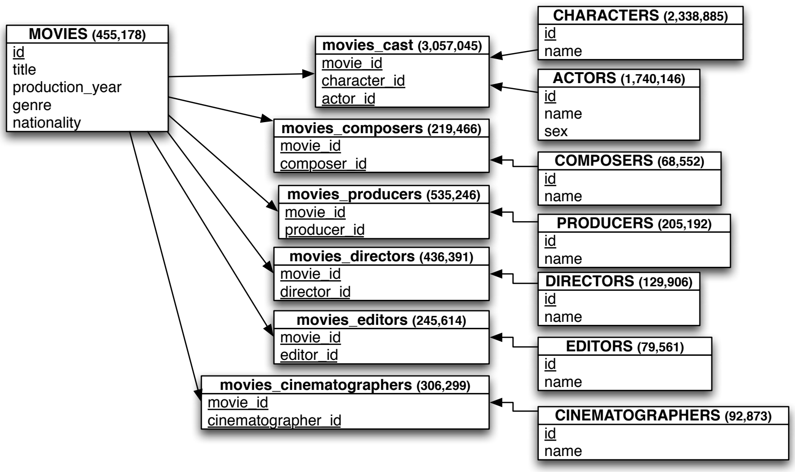

Fig. 2. Schema of IMDb

We use typicality based on strict equality (Formula (2)) since no similarity relations on the domains are available. A conjunction of two criteria is used to define similarity between actors:

1. Director criterion: two actors are considered similar if they have often worked with the same directors;
2. Actor criterion: two actors are considered similar if they have often played with the same other actors.

We have carried out this experimentation using the RDBMS PostgreSQL running on a PC with Intel Core^{TM} Duo CPU T7700 @ 2.40GHz, 2024MB of RAM, a processor cache of 4096 KB, and a hard disk with 16 MB of cache.

5.1 Response Time

Algorithm 1, even modified as explained above (before Example 4), is still rather inefficient inasmuch as it computes the typical sets associated with too many candidate objects (actors, here). In order to improve its performances, we introduced two additional filters:

[1] www.imdb.com

1. Director criterion: one only considers the actors who have worked with at least one of the five most typical directors associated with the target object;
2. Actor criterion: one only considers the actors with which the target actor has played more than once.

We considered twenty target actors and computed the average gain brought by the use of these filters. The results are as follows:

– without any filter, the average response time is 16.18 seconds (2500 objects are evaluated on average), which is obviously unrealistic;
– using both filters, the average response time is 1.51 seconds (138 objects are evaluated), which corresponds to a gain of 90.7%.

5.2 Comparison of the Answer Sets

A second objective was to compare the sets of answers produced by the different matching measures presented in Section 3. We considered Formulas (7), (8), (9), and (12) with $\mu_{most}(x) = x$. Again, we used twenty target actors and considered the top-20 results in each set of answers obtained. We compared every pair of sets of answers $(L_i,\ L_j)$ using the Jaccard indice: $jac(L_i,\ L_j) = \frac{|L_i \cap L_j|}{|L_i \cup L_j|}$. The results are given in Table 1 (each number corresponds to an average computed over the twenty target actors). In this table, L_1, L_2, L_3, and L_4 correspond to the results obtained using Formulas (7), (8), (9), and (12) respectively.

Table 1. Comparison of the sets of top-20 answers (Jaccard indice)

	L_1	L_2	L_3	L_4
L_1	1	0.23	0.29	0.99
L_2		1	0.15	0.23
L_3			1	0.29
L_4				1

It appears that the matching measures produce significantly different results, except for measures (7) and (12) that produce answer sets that are almost always the same: $jac(L_1,\ L_4) = 0.99$. Indeed, we observed that for these two measures, the matching degrees obtained are different, but the sets of top-20 answers are the same, and the elements are even ordered in the same way. Consequently, Formula (12) will not be considered in the subsequent tests.

5.3 Relevance of the Answers

In order to assess the relevance of the answers obtained using the matching measures corresponding to Formulas (7), (8), and (9), we conducted a study involving 16 users who were asked to assess the top-20 results obtained using each of these measures for three target actors (namely Tom Cruise, Julia Roberts, and Robin Williams) corresponding to three individual results that one tries to expand. Again, a conjunction of two criteria ("director" and "actor", cf. above) was used to define the similarity between two actors (but the users were not aware of the similarity criteria used). Each answer could be assessed using one of the three choices "relevant", "not relevant", or "I don't know". For every set of answers L_i and every user u_j, we computed the precision attached to each of the three measures the following way:

$$prec(L_i, u_j) = \frac{|answers\ judged\ \text{“}relevant\text{”}\ in\ L_i\ by\ u_j|}{|answers\ judged\ \text{“}relevant\text{”}\ or\ \text{“}not\ relevant\text{”}\ in\ L_i\ by\ u_j|}.$$

Finally, for each matching measure and each target actor, we computed the average precision over the 16 users. The results appear in Table 2.

Table 2. Average relevance

	L_1	L_2	L_3
Tom Cruise	**0.51**	0.45	0.43
Julia Roberts	**0.47**	0.39	0.13
Robin Williams	0.74	0.78	**0.81**
Average	**0.58**	0.54	0.46

It appears that the matching measure that yields the best precision on average is that based on the Jaccard indice (Formula (7)). It remains to be studied whether a combination of the lists of answers produced by different matching measures could improve the precision of the global result. Let us also mention that the rate of "I don't know" assessments is rather high (between 60 and 75% on average for each of the three measures considered). This also deserves a complementary analysis, in order to evaluate the proportion of such answers that would finally be considered relevant once the user gets familiar with them. Notice that this high rate of "I don't know" assessments does not constitute a weakness of the approach since it is indeed desirable to make the user discover new objects, as in any recommender system.

6 Related Work

Some recent work on keyword search over databases proposes to return "joining networks" of related tuples that together contain a given set of keywords where the tuples are related by foreign key-primary key links [1,3,10]. However, the goal of these approaches is not to retrieve the objects most similar to a given target, but to better cover an initial keyword query.

In [6], the authors consider a class of queries called the "object finder" queries, and their goal is to return the top-k objects that best match a given set of keywords by exploiting the relationships between documents and objects. Contrary to us, the authors consider an information retrieval context where the objects to be retrieved are documents and where the weight of a link between a document and an entity has to be computed by means of a full text search process.

In [11], the authors provide a framework and an engine for the declarative specification of a recommendation process over structured data. The recommendation process in [11] is specified through a series of interconnected operators, which apart from the traditional relational operators, includes also a number of operators specific to the recommendation process, such as the *recommend* operator, that recommends a set of tuples of a specific relation with regards to their relationship with the tuples of another relation. In this approach, the recommendation strategy is rather classical since it is based on similarity between values, not on association between entities.

In [14], the authors study similarity search that is defined among the same type of objects in heterogeneous networks. Intuitively, two objects are similar if they are linked by many paths in the network. Again, this definition is different from ours inasmuch as we do not focus on the number of links between two objects but on the number of entities that are connected to both objects.

In [13], the authors focus on the specific recommendation process of computing YMAL ("You May Also Like") results related to a specific user query. The approach we proposed in this paper clearly belongs to the category of methods that the authors of [13] call "current-state approaches" (where there is no other information available other than a query q posed by a user u and its result $R(q)$), but its originality lies in the exploitation of the notion of association between entities from different tables, which is not explicitly mentioned by the authors of [13].

7 Conclusion

In this paper, we have proposed an approach aimed at retrieving the items similar to a target object from a database, on the basis of the associations that exist between the different entities of the considered database. Intuitively, two objects are similar if they are related with similar sets of entities. Since in general there may exist several links between two given objets, multisets have to be considered, and one makes use of fuzzy set theory in order to i) compute the fuzzy sets of typical entities associated with a given one, ii) compare two fuzzy sets of typical entities. This approach can be seen as the basis of a recommendation mechanism in a structured database context. Of course, it can be used jointly with a value-based similarity approach (objects may be considered similar if they have close values on some attributes *and* are linked to similar sets of entities), which would certainly improve the precision of the result. Preliminary experimental results show that the approach described here is both tractable and promising in terms of relevance/interest of the answers produced.

Among perspectives for future work, we intend to carry out a more complete user study using different databases in diverse applicative contexts (bibliographic database, classified ads database, etc). It is also worth studying how the notion of similarity which makes the most sense for a given type of object could be *learned* or defined *a priori*. Finally, a perspective concerns the definition of a mechanism aimed at providing the user with *explanations* related to the results produced.

References

1. Agrawal, S., Chaudhuri, S., Das, G.: DBXplorer: A system for keyword-based search over relational databases. In: Proc. of ICDE 2002, pp. 5–16 (2002)
2. Andreasen, T., Bulskov, H.: Query expansion by taxonomy. In: Galindo, J. (ed.) Handbook of Research on Fuzzy Information Processing in Databases, pp. 325–349. Information Science Reference, Hershey (2008)
3. Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., Sudarshan, S.: Keyword searching and browsing in databases using BANKS. In: Proc. of ICDE 2002, pp. 431–440 (2002)
4. Bosc, P., Pivert, O.: On the comparison of imprecise values in fuzzy databases. In: Proc. of the 6th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 1997), Barcelona, Spain, pp. 707–712 (1997)
5. Bouchon-Meunier, B., Coletti, G., Lesot, M.-J., Rifqi, M.: Towards a conscious choice of a fuzzy similarity measure: A qualitative point of view. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. LNCS, vol. 6178, pp. 1–10. Springer, Heidelberg (2010)
6. Chakrabarti, K., Ganti, V., Han, J., Xin, D.: Ranking objects based on relationships. In: Proc. of SIGMOD 2006, pp. 371–382 (2006)
7. Dubois, D., Prade, H.: Weighted minimum and maximum operations in fuzzy set theory. Information Sciences 39, 205–210 (1986)
8. Dubois, D., Prade, H.: On data summarization with fuzzy sets. In: Proc. of IFSA 1993, pp. 465–468 (1993)
9. Gaasterland, T.: Relaxation as a platform for cooperative answering. Journal of Intelligent Information Systems 1(3-4), 296–321 (1992)
10. Hristidis, V., Papakonstantinou, Y.: DISCOVER: Keyword search in relational databases. In: Proc. of VLDB 2002, pp. 670–681 (2002)
11. Koutrika, G., Bercovitz, B., Garcia-Molina, H.: Flexrecs: expressing and combining flexible recommendations. In: Proc. of SIGMOD 2009, pp. 745–758 (2009)
12. Pappis, C., Karacapilidis, N.: A comparative assessment of measures of similarity of fuzzy values. Fuzzy Sets and Systems (1993)
13. Stefanidis, K., Drosou, M., Pitoura, E.: You may also like" results in relational databases. In: Proc. of PersDB 2009 (2009)
14. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. PVLDB 4(11), 992–1003 (2011)
15. Yager, R.: General multiple-objective decision functions and linguistically quantified statements. International Journal of Man-Machine Studies 21(5), 389–400 (1984)
16. Yager, R.: Interpreting linguistically quantified propositions. International Journal of Intelligent Systems 9(6), 541–569 (1994)
17. Zadeh, L.: A computational approach to fuzzy quantifiers in natural languages. Computing and Mathematics with Applications 9, 149–183 (1983)
18. Zadeh, L.: A computational theory of dispositions. International Journal of Intelligent Systems 2, 39–63 (1987)