

# Qualitative pattern matching with linguistic terms

Yannick LOISEAU, Henri PRADE

*Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier*

Email: {loiseau,prade}@irit.fr

**Abstract.** In the framework of possibility theory, a tool named ‘fuzzy pattern matching’ (FPM) has been proposed in the eighties and since successfully used in flexible querying of fuzzy databases and in classification. Given a pattern representing a request expressed in terms of fuzzy sets, and a database storing imprecise or fuzzy attribute values for some data, the FPM returns two matching degrees. Namely, for each item in the base, the possibility and the certainty that it matches the requirements of the pattern are computed. In multiple-source information systems, attributes values are often assessed in linguistic terms belonging to different vocabularies. The request itself, which may include preferences, may be expressed using terms of another vocabulary. The paper proposes a counterpart of FPM, called ‘Qualitative Pattern Matching’ (QPM), for estimating levels of matching between a request and data expressed with words; words can be related together through a qualitative thesaurus or ontology, where approximate synonymy and specialization relations are encoded. Given a request, QPM rank-orders the items which possibly, or which certainly match the requirements, according to the preferences of the user. The proposed approach does not require any numerical computation of similarity degrees and is qualitative in nature. It is illustrated on an example, and its merits for dealing with information querying in face of heterogeneous sources of information are advocated.

## Introduction

Information, even in standardized form, is often expressed by words as well as numbers. When information come from various sources, the vocabularies used for expressing information are heterogeneous. Categories pertaining to the same concept, used by one source, do not perfectly match categories used by another. Moreover, even in the case of a single source, requests may not be specified exactly in the terms used in the data base, which may not be known by users. The problem of the heterogeneity of the sources may be also due to the use of multilingual information sources.

Measures of semantic similarity between words have been thoroughly studied in the information retrieval literature, taking advantage of distances between nodes in a taxonomy, or based on common probabilistic information content (e.g. [1]). One commonly investigated strategy when a user’s query fails, is to generate similar queries in place of it (e.g [2]) on the basis of ontologies or thesaurus. Generally speaking, these concerns may be seen as parts of a new research trend, sometimes referred to as “computing with words” ([3]).

Besides, a querying process may involve user’s preferences which can be taken into account when the queries are allowed to be flexible (e.g. [4]). Then, the pieces of information

which are retrieved are rank-ordered according to the user's preferences. Fuzzy set based approaches have been developed for representing flexible queries, and can be applied to regular databases as well as fuzzy databases containing ill-known attribute values, also represented by means of fuzzy sets. A tool, called 'fuzzy pattern matching' ([5, 6, 7]) has been proposed in the framework of possibility theory, which computes to what extent it is possible, and to what extent it is certain that a piece of information, encoded as a tuple of (fuzzily) known attribute values, satisfies a flexible request expressed by means of fuzzy sets representing the preference profiles of the user on the attributes of interest.

In fuzzy pattern matching, each label appearing in the request or in the database is represented by a fuzzy set. Fuzzy sets defined on the same attribute domain can be compared, by means of a set of two measures, which acknowledges the asymmetry between the pattern which expresses a requirement and the pieces of information. In this paper, we keep the main features of the fuzzy pattern matching approach as much as possible, and we adapt it to symbolic labels. The intended purpose of the approach is to deal with queries stated in terms of linguistic labels (may be weighted for expressing preferences). These queries are to be evaluated in face of a database also containing linguistic terms. The matching between the labels in the request and the data does not require perfect identity, but will be a matter of semantic similarity computed by means of a weighted network associated with each attribute domain. Thus, the labels are no longer explicitly associated with fuzzy set representations, but their semantic relationships are still assumed to be estimated in terms of two measures, as in the fuzzy pattern matching technique, and the evaluation process remains qualitative.

The paper is organised as follows. Section 1 provides a background on fuzzy pattern matching. Section 2 states the qualitative pattern matching problem. Section 3 presents the approach and illustrates it on an example and Section 4 discusses the main features of the approach and concludes.

## 1 Background on fuzzy pattern matching

By pattern, we mean here a set of elementary requirement encoding by labels of properties referring to attribute domain. The basic idea is to attach to each label of a pattern the membership function of a fuzzy set restricting the values which are more or less compatible with the meaning of the label. These values belong to some prescribed domain corresponding to the range of the attribute which the label refers to. For instance, the label *tall* refers to a scale of heights, and corresponds to a fuzzy subset of this domain. In place of a numerical domain, we may have a discrete set of typical elements as well. Besides, data are also represented by lists of labels whose components are associated with fuzzy sets. These fuzzy sets are viewed as possibility distributions which model the imprecision pervading the data, and restrict the more or less possible values of the considered attributes. Such lists contain possibly ill-known attribute values pertaining to the description of objects. Namely, a component in a list refers to only one (ill-located) element of the domain of the concerned attribute (which is supposed to be single-valued).

The basic asymmetry of the pattern-data matching is preserved by this modeling convention. Indeed, a fuzzy pattern represents an imprecisely described class of objects which are looked for. Namely, let  $T$  and  $T'$  be respectively a pattern label (i.e. a requirement) and an item component pertaining to the same single-valued attribute (i.e. a piece of data), which are to be compared.  $T$  and  $T'$  refer to the same domain  $U$  conveying their meanings. Let  $\mu_T$  be

the membership function associated to label  $T$  and  $\pi_{T'}$  be the possibility distribution attached to  $T'$ . Both are mappings from  $U$  to  $[0, 1]$ . Let  $u$  be an element of  $U$ . Then  $\mu_T(u)$  is the grade of compatibility between the value  $u$  and the meaning of  $T$ . Namely,  $\mu_T(u) = 1$  means total compatibility with  $T$  and  $\mu_T(u) = 0$  means total incompatibility with  $T$ . By contrast,  $\pi_{T'}(u)$  is the grade of possibility that  $u$  is the value of the attribute describing the object modelled by the item.  $T'$  is a fuzzy set of *possible* values (only one of which is the genuine value of the ill-known attribute), while  $T$  is a fuzzy set of *more or less* compatible values. For instance,  $\pi_{T'}(u) = 1$  means that  $u$  is totally possible (there may exist distinct values  $u$  and  $u'$  such as  $\pi_{T'}(u) = \pi_{T'}(u') = 1$ ), while  $\pi_{T'}(u) = 0$  means that  $u$  is totally impossible as an attribute value of the object to which the item pertains. In the following,  $\mu_T$  and  $\pi_{T'}$  are always supposed to be normalised, i.e. there is always a value which is totally compatible with  $T$ , and a value totally possible in the range  $T'$ .

Two scalar measures are used in order to estimate the compatibility between a request element (pattern atom)  $T$  and its counterpart  $T'$  in the data attribute (item list), namely a degree of possibility of matching  $\Pi(T; T')$  and a degree of necessity of matching  $N(T; T')$  which are respectively defined by ([7]):

$$\Pi(T; T') = \sup_{u \in U} \min(\mu_T(u), \pi_{T'}(u)),$$

$$N(T; T') = \inf_{u \in U} \max(\mu_T(u), 1 - \pi_{T'}(u)).$$

The limiting cases where  $\Pi(T; T')$  and  $N(T; T')$  take values 0 and 1 are useful to study in order to lay bare the semantics of these indices. For any fuzzy set,  $F$  on  $U$ , let  $F^\circ = \{u \in U \mid \mu_F(u) = 1\}$  be the core of  $F$ , and  $s(F) = \{u \in U \mid \mu_F(u) > 0\}$  its support. Then it can be checked that:

1.  $\Pi(T; T') = 0$  if and only if  $s(T) \cap s(T') = \emptyset$ ,
2.  $\Pi(T; T') = 1$  if and only if  $T^\circ \cap T'^\circ \neq \emptyset$ ,
3.  $N(T; T') = 1$  if and only if  $s(T') \subseteq T^\circ$ ,
4.  $N(T; T') > 0$  if and only if  $T'^\circ \subset s(T)$  (strict inclusion).

The measure  $\Pi(T; T')$  estimates to what extent it is possible that  $T$  and  $T'$  refer to the same value  $u$ , in other words,  $\Pi(T; T')$  is a degree of *overlapping* of the fuzzy set of values compatible with  $T$ , with the fuzzy set of possible values of  $T'$ . The measure  $N(T; T')$  estimates to what extent it is necessary (i.e. certain) that the value to which  $T'$  refers is among the ones compatible with  $T$ ; in other words,  $N(T; T')$  is a degree of *inclusion* of the set of possible values of  $T'$  into the set of values compatible with  $T$ .

It can be shown that  $\Pi(T; T') \geq N(T; T')$ . Note that when  $T'$  is precise, i.e.  $\exists t', \pi_{T'}(t') = 1$  and  $\forall u \neq t', \pi_{T'}(u) = 0$  which can be written  $T' = \{t'\}$ , then

$$\Pi(T; \{t'\}) = N(T; \{t'\}) = \mu_T(t')$$

The atomic measures of possibility and necessity are aggregated separately in order to obtain two global measures between the whole pattern and the whole item. When the pattern expresses a conjunction of elementary requirement " $T_1$  and  $\dots T_n$ ", this aggregation is performed using the min operation and preserves the respective semantics of the measures in terms of possibility and necessity. Indeed, we have ([7]):

$$\Pi(T_1 \times \dots \times T_n; T'_1 \times \dots \times T'_n) = \min_{i=1, \dots, n} \Pi(T_i; T'_i) \quad (1)$$

$$N(T_1 \times \dots \times T_n; T'_1 \times \dots \times T'_n) = \min_{i=1, \dots, n} N(T_i; T'_i) \quad (2)$$

where  $T_i$  and  $T'_i$  are supposed to be defined on the same domain  $U_i$ , and where  $\times$  denotes the Cartesian product defined for two fuzzy sets  $F_i$  and  $F_j$  by :

$$\forall u_i \in U_i, \forall u_j \in U_j, \mu_{F_i \times F_j}(u_i, u_j) = \min(\mu_{F_i}(u_i), \mu_{F_j}(u_j)).$$

## 2 The symbolic matching problem

We still assume now that the labels which are used, refer to precisely identified attributes. This means that the items stored in the databases are described in terms of attributes  $i$ , with  $i = 1, n$ . For each attribute  $i$ , let  $\mathcal{T}_i$  be the set of labels pertaining to it. Namely,  $\mathcal{T}_i = \{t_{ij}, j = 1, n(i)\}$  where  $t_{ij}$  denotes a label which can be used for assessing the value of attribute  $i$ . Labels pertaining to the same attribute are no longer associated with fuzzy set representations as already said, but their meanings are related through a so-called ‘‘possibilistic ontology’’  $O_i$  ([8]).

This means that  $O_i$  is associated with two graded relations. Namely, for two labels  $t_{ij}$  and  $t_{ik}$  we have:

- $\Pi(t_{ij}, t_{ik}) = \Pi(t_{ik}, t_{ij})$  assesses to what extent  $t_{ij}$  and  $t_{ik}$  can refer to the same thing. Note that  $\Pi(t_{ij}, t_{ik}) = 0$  means that the two labels never refer to the same thing.
- $N(t_{ij}, t_{ik})$  assesses to what extent it is certain that  $t_{ik}$  is a specialization of  $t_{ij}$ .  $N$  is not symmetrical.  $N(t_{ij}, t_{ik}) = 1 = N(t_{ik}, t_{ij})$  expresses that  $t_{ij}$  and  $t_{ik}$  are perfectly synonymous.  $N(t_{ij}, t_{ik}) = 0$  expresses a total lack of certainty that  $t_{ik}$  is a specialization of  $t_{ij}$ .

The possibly graded relations  $\Pi$  and  $N$  are defined on a subset of the Cartesian product  $\mathcal{T}_i \times \mathcal{T}_i$ , for each  $i$  as it will be seen on an example in the next section. The relations are supposed to be completed by taking advantage of the following properties:

$$N(t_{ij}, t_{ih}) \geq \min(N(t_{ij}, t_{ik}), N(t_{ik}, t_{ih})) \quad (3)$$

This expresses the transitivity of the specialization (see [9]).

$$\Pi(t_{ij}, t_{ih}) \geq \min(N(t_{ij}, t_{ik}), \Pi(t_{ik}, t_{ih})) \quad (4)$$

This expresses that if  $t_{ik}$  specializes  $t_{ij}$  and if  $t_{ik}$  and  $t_{ih}$  can refer to the same thing, then the meanings of  $t_{ij}$  and  $t_{ih}$  overlaps as well (since  $t_{ij}$  encompasses a larger set of situations than  $t_{ik}$ ).

Moreover it is clear that  $\Pi(t_{ij}, t_{ij}) = N(t_{ij}, t_{ij}) = 1$ , and  $\Pi(t_{ij}, t_{ik}) = \Pi(t_{ik}, t_{ij})$ . It is assumed that  $\Pi(t_{ij}, t_{ik}) \geq N(t_{ij}, t_{ik})$ , since specialization entails that the meanings overlap. In the same way, if  $N(t_{ij}, t_{ik}) > 0$  we should have  $\Pi(t_{ij}, t_{ik}) = 1$ , since if it is somewhat necessary (i.e. certain) that  $t_{ik}$  matches  $t_{ij}$ , it has to be fully possible.

So in practice, starting with a partial definition of  $N$  and  $\Pi$ , the relations are completed by applying repeatedly (3) and (4) and the above constraints. The other non specified values of  $N$  or  $\Pi$  will be assumed to be zero by default.

In the following, we still use binary-valued measures  $\Pi$  and  $N$  first. In any cases, only a small number of intermediary grades between 1 and 0 will be used. It will enables us to distinguish in particular between situations where a term is always a specialization of another ( $N(t_{ij}, t_{ik}) = 1$ ), from situations where it is *generally* a specialization ( $N(t_{ij}, t_{ik}) > 0$ ).

Requests will be also stated in terms of  $t_{ij}$ 's belonging to the  $\mathcal{T}_i$ 's. A request  $R$  will be seen as a set  $\{T_i\}$  representing a conjunction of elementary requirements  $T_i$  where each  $T_i$  will be a disjunction of  $t_{ij}$ 's where  $t_{ij} \in \mathcal{T}_i$ , namely  $T_i = \bigvee_{j \in R(T_i)} t_{ij}$  where  $R(T_i)$  is the set of indices involved in  $T_i$ . More generally, the disjunction will be prioritised, in order to express user's preferences.

It will be first assumed that the attribute values of data are described by means of a unique label,  $T'_i = \{t'_{ik}\}$  where  $t'_{ik} \in \mathcal{T}_i$ . More generally,  $T'_i$  will not be represented by a singleton, but by a disjunction of  $t'_{ik}$ 's for expressing that the attribute value is imprecisely known with respect to the vocabulary  $\mathcal{T}_i$ .

### 3 Presentation of the proposed approach and illustrative example

#### 3.1 A "possibilistic ontology"

To illustrate the approach, we will consider a database made of holidays' places, with only three attributes to keep the example simple enough. Attributes are:

1. The lodging type, which is a name like *hotel* or *campsite*, (or more generally a disjunction)
2. The place (country, areas, etc.)
3. The price, which is a numeric value (which will be briefly considered only at the end of the paper).

Possibilistic ontologies, in the sense of section 2 are used to define a vocabulary pertaining to each attribute. Considering that data have  $n$  linguistic attributes, we would need to define  $n$  ontologies. However, numerical attributes don't need a restricted vocabulary, as the price in our case. Let  $\Omega$  be the set of ontologies we will use:  $\Omega = \{O_i\} \text{ } i=1, n$ .

Each ontology  $O_i$  is composed of terms  $t_{ij}$ :  $\forall i \in \llbracket 1; n \rrbracket, O_i = \{t_{ij}, j = 1, n(i)\}$ , where  $n(i)$  is the number of possible terms for attribute  $i$ .

As already said in section 2, we use possibility ( $\Pi$ ) and necessity ( $N$ ) to represent synonymy and specialisation relations in ontologies. For example, if  $t_{ij}$  is an hyponym of  $t_{ik}$  (i.e.  $t_{ij}$  specializes  $t_{ik}$ ), then  $N(t_{ik}; t_{ij}) = 1$ . Otherwise, if  $0 < N(t_{ik}; t_{ij}) \leq 1$ , we are not completely sure that the term  $t_{ij}$  is more specific than  $t_{ik}$ .

Here, we define two ontologies, one for the *lodging types* (see Fig. 1) and one for *places* (Fig. 2). These networks are simplified representations, since complete ontologies would be huge and can be recovered using (3) and (4) and the other constraints as explained in Section 2.

Note that some words like *lodge* and *inn*, or *motel* and *motor inn* are considered as *possible* synonyms. However, note that assessing for *lodge* and *inn*, that the possibility is 1, gives

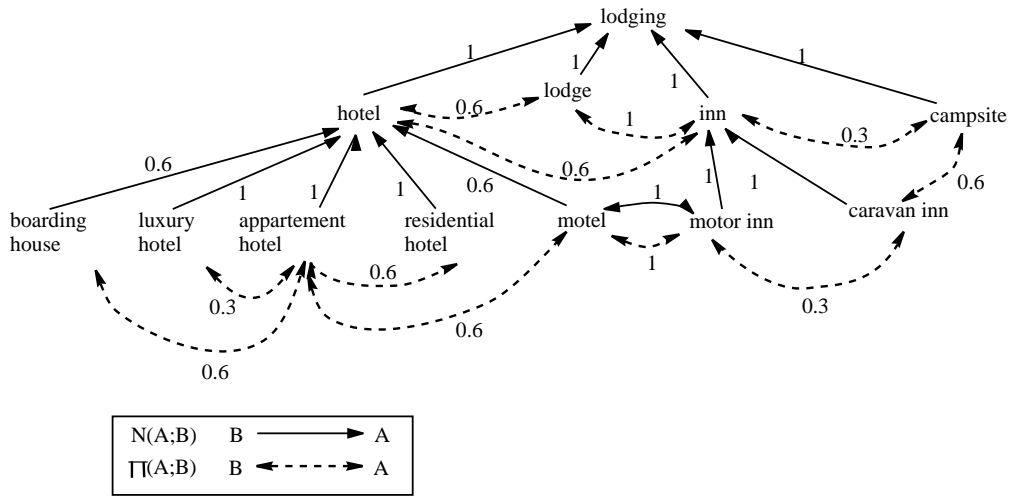


Figure 1: Ontology defining the lodging vocabulary

no information concerning the necessity. It is possible for some *lodges* to be an *inn*, and vice-versa, but it could exist *lodges* that are not *inns*. Besides, when the necessity between two terms, as for *motel* and *motor inn*, is "1", the two terms are known as genuine synonyms, that is, they have exactly the same meaning.

Values of possibilities and necessities in such an ontology are qualitative in nature, and determined by the semantics of the terms. For example,  $N(\text{hotel}; \text{motel}) = 0.6$ , means that we suppose that it exists motels that cannot be considered as hotels. Although, we are using a numerical encoding, the values are not meaningful in themselves, but just their orderings. In practice, we use only a small number of possible values, here  $\{0, 0.3, 0.6, 1\}$ .

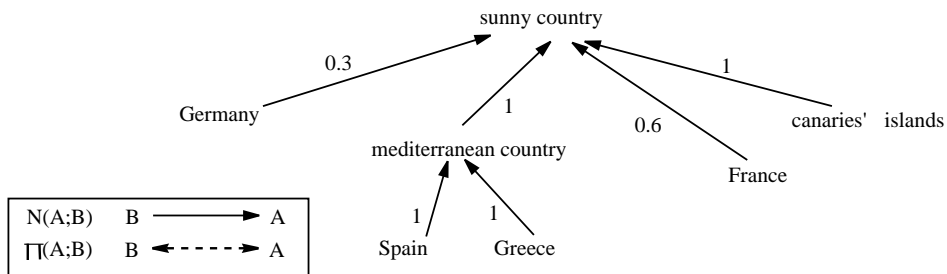


Figure 2: Ontology defining the places vocabulary

### 3.2 Evaluation of a simple request

A query is described as a set  $R$  of terms or compound terms  $T_i$  for a set of attributes. There is almost one  $T_i$  in  $R$  for each attribute  $i$ . The set  $R$  will be interpreted as a conjunction in the evaluation. Namely,

$$R = \bigwedge_{i \in A(R)} T_i$$

where  $A(R)$  is the set of attributes involved in the query.

Each  $T_i$  is a disjunction of terms from the vocabulary defined by the ontology associated with attribute.

$$T_i = \bigvee_{j \in R(T_i)} t_{ij}$$

where  $R(T_i)$  is the set of the terms involved in  $R$  for attribute  $i$ . Each  $t_{ij}$  belongs to the ontology  $O_i$ .

To evaluate a query means to retrieve all data  $T'$  such that  $\Pi(R, T')$  or  $N(R, T')$  are non zero, where  $\Pi(R, T')$  and  $N(R, T')$  estimate to what extent the piece of data  $T'$  possibly or certainly matches the request  $R$ . Formally, we have to evaluate  $\pi_i = \max_{j \in R(T_i)} \Pi(t_{ij}, t'_{ik})$ . The max operator is used since we have defined  $T_i$  as a disjunction. Likewise, we compute the necessity value  $\nu_i = \max_{j \in R(T_i)} N(t_{ij}, t'_{ij})$ . Note that if the data term for attribute  $i$   $t'_{ij}$  is the same as the query attribute  $t_{ij}$ , then  $\pi_i = \nu_i = 1$  and the piece of data matches exactly the query for the  $i^{th}$  attribute.

Since the query is supposed to be a conjunction over attributes, we compute the final score of the query as the min between attributes.

$$\Pi(R, T') = \min_{i=1, n} \pi_i$$

$$N(R, T') = \min_{i=1, n} \nu_i$$

Then, the piece of data  $T'$  are sorted first according to the decreasing values of  $N(R, T')$  and then according to the decreasing values of  $\Pi(R, T')$  for  $T'$  sharing the same value of  $N(R, T')$ .

In this section, the above disjunction is supposed to be a crisp one, but we will extend it to a fuzzy one in the next subsection. An example of a query can be:

$$R = (hotel \vee inn) \wedge (sunnycountry)$$

In the following, we only consider the attributes involved in the query, for the pieces of data. Like in the query, data are sets of terms or compound terms for the considered attributes, namely  $T' = \{T'_i, i \in A(R)\}$ . In this section, we assume that each data attribute values are singletons of terms from the ontology, i.e.  $T'_i = \{t'_{ik}\}$  where  $t'_{ik} \in O_i$ .

For instance, let's consider the above query  $R = (hotel \vee inn) \wedge (sunnycountry)$ . Let's assume we have the data base:

	lodging	place
1	hotel	England
2	boarding house	Spain
3	lodge	Greece
4	motel	France

Let's evaluate the query. For the first row, we have  $\pi_{lodging} = \max(\Pi(hotel, hotel), \Pi(inn, hotel))$  and  $\pi_{place} = \Pi(sunnycountry, England)$ . Obviously,  $\Pi(hotel, hotel) = 1$  and according to the ontology,  $\Pi(inn, hotel) = 0.6$ , so  $\pi_{lodging} = 1$ . As *England* has no relation with *sunny country*, we get that  $\pi_{place} = 0$ . So  $\Pi(R, T'_1) = \min(\pi_{lodging}, \pi_{place}) = 0$ .

Likewise, for the second row, we have  $\nu_{lodging} = \max(N(\text{hotel}, \text{boardinghouse}), N(\text{inn}, \text{boardinghouse}))$ . With the ontology, we have  $N(\text{hotel}, \text{boardinghouse}) = 0.6$  and  $N(\text{inn}, \text{boardinghouse}) = 0$ , so  $\nu_{lodging} = 0.6$ . In the same way,  $\nu_{place} = N(\text{sunnycountry}, \text{Spain}) = 1$ . Consequently,  $N(R, T'_2) = \min(\nu_{lodging}, \nu_{place}) = 0.6$

We can check that  $N(R, T'_3) = 0$  and  $\Pi(R, T'_3) = 1$ .

The fourth row illustrates a “transitivity-like” property. Whereas we have no information about  $N(\text{inn}, \text{motel})$ , the ontology gives  $N(\text{inn}, \text{motorinn}) = 1$  and  $N(\text{motorinn}, \text{motel}) = N(\text{motel}, \text{motorinn}) = 1$ . Namely, we know that a *motorinn* is a specialisation of *inn* and that *motel* and *motorinn* are synonymous. We can infer that  $N(\text{inn}, \text{motel}) = 1$ . In the same way, we have  $N(\text{hotel}, \text{motorinn}) = 0.6$  since  $N(\text{hotel}, \text{motel}) = 0.6$ . Finally, we get :  $N(R, T'_4) = \min(0.6, 0.6) = 0.6$  using the ontology pictured in Fig.2.

Note that the “transitivity” is only allowed in the sense of (3) and (4). But for example, we could not infer anything about  $\Pi(\text{lodge}, \text{motorinn})$  from  $\Pi(\text{lodge}, \text{inn}) = 1$  and  $N(\text{inn}, \text{motorinn}) = 1$ . Indeed, we know that *lodge* and *inn* intersects and that *motorinn* is a specification of *inn*, but it can correspond to a type of *inns* that are not *lodges*.

As an answer, we obtain the following ranking: 2, 4 and then 3. Indeed row no.4 is ranked after the no.2, since no.4 has obtained the grade 0.6 for necessity on both criteria (lodging and place), while no.2 has a better grade on one of the query’s criteria ( $\nu_{place} = 1$ ).

Row	$\Pi$	N	Rank
1	0	0	4
2	1	0.6	1
3	1	0	3
4	1	0.6	2

Table 1: Result for the query

### 3.3 Prioritised requests

Let’s consider a more general query by allowing *fuzzy* disjunction. This enables the user to express preferences in the data selection, by prioritising the query terms.

This can be done by adding a weight to each query term. The query element  $T_i$  is now considered as a fuzzy set, and the weights represent to what extent the term is belonging to the query. Let  $\lambda_{ij}$  be the weight of the term  $t_{ij}$ . We can write symbolically  $T_i = \bigvee_{j \in R(T_i)} \lambda_{ij} / t_{ij}$ , where  $\lambda_{ij}$  is the weight of term  $t_{ij}$ . We assume that  $\lambda_{ij} \in [0, 1]$  and that  $\max_j \lambda_{ij} = 1$ , which means that there is at least one term that fits the user’s need.

This representation can be used to express ideas or concepts that are not defined in the ontology, by combining different existing terms. For example, the user can define a *cosy lodging* as :  $\{(0.5, \text{lodge}); (0.7, \text{motel}); (0.85, \text{apartmenthotel}); (1, \text{luxuryhotel})\}$

The query is then evaluated in the same way. We have now ([7]):

$$\pi_i = \max_{j \in R(T_i)} (\min(\lambda_{ij}, \Pi(t_{ij}, t'_{ik}))),$$

$$\nu_i = \max_{j \in R(T_i)} (\min(\lambda_{ij}, N(t_{ij}, t'_{ik}))).$$

Let’s consider:  $R = (\text{cosylodging}) \wedge (\text{sunnycountry})$ . For the fourth row,  $\nu_{lodging}$  is no more 0.6. We have  $\nu_{lodging} = \max(\min(0.7, 1), \min(0.85, 0.6)) = 0.7$

The same kind of weight can be used to express a preference between attributes themselves. In our example, the *lodging type* can be less important than the country, so that the query become:  $R = \bigwedge_{i \in R(i)} (\omega_i, T_i)$  where  $\omega_i$  has the same constraints as  $\lambda_{ij}$  and represent the importance of the attribute in the query. Here, as we have a conjunction, the evaluation will be ([7]):

$$\begin{aligned}\Pi(R, T') &= \min_{i \in R(i)} (\max(1 - \omega_i, \pi_i)), \\ N(R, T') &= \min_{i \in R(i)} (\max(1 - \omega_i, \nu_i)).\end{aligned}$$

### 3.4 Disjunctive data

To be more general, we can allow for disjunctive labels in the data, i.e. imprecise descriptions such as *hotel*  $\vee$  *inn*. For each attribute,  $T'_i$ , which was a singleton, can be now a set, or more generally a fuzzy set of terms. We now have  $T'_i = \bigvee_{k \in D(T'_i)} \lambda'_{ik} / t'_{ik}$ , where  $D(T'_i)$  is the set of terms involved in the attribute value of the piece of data. The evaluation of possibility and necessity degrees becomes:

$$\begin{aligned}\Pi(T_i, T'_i) &= \max_{j,k} \min (\Pi(t_{ij}, t'_{ik}), \lambda_{ij}, \lambda'_{ik}) \\ N(T_i, T'_i) &= \max_{j \in R(T_i)} \min \left( \lambda_{ij}, \min_{k \in D(T'_i)} \max(1 - \lambda'_{ik}, N(t_{ij}, t'_{ik})) \right)\end{aligned}$$

The formula giving  $N(T_i, T'_i)$  expresses that it should exist a term  $t_{ij}$  in the request such that all the  $t'_{ik}$ 's appearing in  $T'_i$  are specializations of  $t_{ij}$ . Indeed, the description of the attribute value of piece of the data is imprecise and whatever the attribute value is, we should be certain that the request is satisfied. Moreover, the requirement that  $t'_{ik}$  is a specialisation of  $t_{ij}$  is all the less compulsory as  $\lambda'_{ik}$  is small (at the extreme, when  $\lambda'_{ik} = 0$ , i.e.  $t'_{ik}$  does not appear in  $T'_i$ ,  $N(t_{ij}, t'_{ik})$  should have no influence.

An example of such a data is:  $D = \{(1, \textit{hotel}); (0.5, \textit{motel})\}; \textit{France}$ , which means that it is *possibly* an hotel or a motel, and more likely an hotel. Let the query be  $R = (1; \textit{hotel})$ . The ontology gives  $N(\textit{hotel}, \textit{motel}) = 0.6$  and  $\Pi(\textit{hotel}, \textit{motel}) = 1$ . Thus, we have:

$$\begin{aligned}\pi_{\textit{lodging}} &= \max(\min(1, 1, 1), \min(1, 1, 0.5)) = 1 \\ \nu_{\textit{lodging}} &= \min(1, \min(\max(1 - 1, 1), \max(1 - 0.5, 0.6))) = 0.6\end{aligned}$$

Indeed the attribute value can be a *motel* and a *motel* can differ from an *hotel* (see Fig.1). As we are looking for an *hotel*, the data does not match perfectly the request, which is shown by the necessity  $\nu_{\textit{lodging}} < 1$ .

Note that if the data attribute is a disjunction of perfect synonyms ( $t'_{i1} \vee t'_{i2}$ ) having the same weight, the evaluation will be equivalent has considering only one of the terms. Indeed  $\Pi(t'_{i1}, t'_{i2}) = 1$  and  $N(t'_{i1}, t'_{i2}) = N(t'_{i2}, t'_{i1}) = 1$ , since the two terms are perfectly matching, and thus it can be checked that  $N(t_{ij}, t'_{i1}) = N(t_{ij}, t'_{i2})$ , using (3).

## 4 Concluding remarks

The paper has proposed a new approach for dealing with linguistic terms in querying systems. The main features of the approach are:

- The relations between terms  $t_i$  and  $t_k$  are reflected by means of two indexes.  $\Pi(t_i, t_k)$  assesses to what extent  $t_i$  and  $t_k$  can refer to the same thing, while  $N(t_i, t_k)$  evaluates to what extent it is certain that  $t_k$  specializes  $t_i$ . Note that here, the similarity between terms does not depend on any hierarchical distance between nodes in a taxonomy tree.
- There is no need to reformulate a query using similar terms if the initial query fails, as it is the case with classical approaches, since an exact matching of the terms of the query is not required.
- Preferences can be represented in the query by weighting the terms used in it. The importance of the attributes can be assessed as well.
- The data themselves may be imprecise since the description of an attribute value can be made by a fuzzy set of terms.
- The evaluation in qualitative pattern matching parallels the one made by fuzzy pattern matching. So it enables us to mix linguistic terms and numbers in the data. Thus in our example, if we want also to take into account some requirement about the price, for each item in the database, we compute to what extent it is possible and certain that the price agrees with requirement, and we aggregate these evaluations conjunctively with the corresponding evaluations pertaining to the other attributes.

## References

- [1] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problem of ambiguity in natural language. *J. of Art. Int. Res.*, 1998.
- [2] A. Bidault, C. Froidevaux, and B. Safar. Proximité entre requêtes dans un contexte médiateur. *RFIA 2002*, 2:653–662, 2002.
- [3] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.
- [4] T. Andreasen, H. Christiansen, and H. L. Larsen. *Flexible Query Answering Systems*. Kluwer Academic Publishers, 1997.
- [5] M. Cayrol, H. Farreny, and H. Prade. Fuzzy pattern matching. *Kybern.*, 11:103–16, 1982.
- [6] D. Dubois, H. Prade, and C. Testemale. Weighted fuzzy pattern matching. *Fuzzy Sets and Systems*, 28:313–331, 1988.
- [7] D. Dubois and H. Prade. Tolerant fuzzy pattern matching: An introduction. In P. Bosc and J. Kacprzyk, editors, *Fuzziness in Database Management Systems*, pages 42–58. Physica-Verlag, 1995.
- [8] H. Farreny and H. Prade. Dealing with vagueness of natural languages in man-machine communication. In W. Karwowski and A. Mital, editors, *Applications of Fuzzy Set Theory in Human Factors*, pages 71–85. Elsevier, 1986.
- [9] J.P. Rossazza, D. Dubois, and H. Prade. A hierarchical model of fuzzy classes. In R. De Caluwe, editor, *Fuzzy and Uncertain Object-Oriented Databases*, pages 21–62. World Pub. Co., 1997.