

Qualitative pattern matching with linguistic terms

Yannick LOISEAU, Henri PRADE,
Mohand BOUGHANEM

*IRIT, Université Paul Sabatier, 118 route de
Narbonne, 31062 Toulouse Cedex 4, France
Email: {loiseau,prade,boughane}@irit.fr*

In the setting of possibility theory, a tool named ‘fuzzy pattern matching’ (FPM) has been proposed in the eighties, and then 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 containing imprecise or fuzzy attribute values, 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, attribute 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 is based on a qualitative assessment of matching degrees which does not necessarily require the use of numerical scales. Its merits for dealing with information querying in face of heterogeneous sources of information are advocated. Application to the handling of textual data in information retrieval is also outlined.

Keywords: pattern matching, possibility theory, similarity, preference, information systems.

1. Introduction

Information, even in standardized form, is often expressed by words as well as numbers. When information comes from various sources, the vocab-

ularies used for expressing information are heterogeneous. Categories pertaining to the same concept, used by one source, do not perfectly match categories used by another in general. Moreover, even in the case of a single information source, requests may not be specified exactly in the terms used in the data base, since they 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. [14]). 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” [16].

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. [1, 11]). 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’ [5,9,8] has been proposed in the framework of possibility theory. It 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 FPM, 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 (maybe 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 net 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 in nature.

The paper, a fully revised version of [12], is organized as follows. Section 2 provides a background on fuzzy pattern matching. Section 3 states the qualitative pattern matching problem. Sections 4 to 7 present the approach and illustrates it on a running example. The notion of “possibilistic ontology” is first introduced. Based on this notion, the evaluation of a request is discussed, before considering requests including priority levels, or allowing for disjunctive data in the information source. Section 8 outlines an application of the approach to the handling of textual information. Section 9 summarizes the main features of the approach.

2. Background on fuzzy pattern matching

By a pattern, we mean here a set of elementary requirements encoded by labels of properties referring to attribute domains. For instance, the pattern ‘*cheap* and *large*’ in face of a database storing descriptions of houses to let is supposed to specify the requirement that the value of the attributes, price and size, for the houses that the user is in search of, should respectively match with ‘*cheap*’ and ‘*large*’. 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. Thus, in our example, ‘*cheap*’ and ‘*large*’ are associated with membership functions defined on the respective attributes domains. In place of a numerical domain, we may have a discrete set of typical values 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, which may be ill-known. Namely, the possibility distribution, corresponding to a label in the requirement 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 μ_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 associated with 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. In particular, $\pi_{T'}(u) = 1$ means that u is totally possible (however, 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, μ_T and $\pi_{T'}$ are always supposed to be normalized, 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 (item list): a degree of possibility $\Pi(T; T')$ and a degree of necessity $N(T; T')$ defined by [5]:

$$\Pi(T; T') = \sup_{u \in U} \min(\mu_T(u), \pi_{T'}(u)), \quad (1)$$

$$N(T; T') = \inf_{u \in U} \max(\mu_T(u), 1 - \pi_{T'}(u)). \quad (2)$$

The measure $\Pi(T; T')$ estimates to what extent it is possible that T and T' refer to the same value u . $\Pi(T; T')$ 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 . $N(T; T')$ estimates the *inclusion* of the possible values of T' into the set of values compatible with T .

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 [9]:

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

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

Note also that for $\mu_T = \pi_{T'}$, $\Pi(T; T') = 1$, and when T is a genuine fuzzy set, $1 > N(T; T') \geq \frac{1}{2}$. This acknowledges the fact that even if $\mu_T = \pi_{T'}$, we cannot be completely certain that a value restricted by the possibility distribution $\pi_{T'}$ is included in the core of the fuzzy set T . In particular for continuous membership functions on domains which are subsets of the real line, (2) yields $N(T; T') = \frac{1}{2}$ if $\mu_T = \pi_{T'}$. Indeed, there are values which are close to be possible at degree 0.5, and which are not in the 0.5-level cut of T , i.e. in $\{u, \mu_T(u) \geq \frac{1}{2}\}$.

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 requirements “ T_1 and ... and 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 [9]:

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

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

3. The symbolic matching problem

In the following, we still assume that the labels which are used in the request or in the base refer to precisely identified attributes. This means that the items stored in the database 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, $\mathcal{T}_i = \{t_i^j, j = 1, n(i)\}$, where t_i^j denotes a label (e.g. ‘hotel’) which can be used for assessing the value of attribute i (e.g. ‘lodging’). 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 , following an idea already suggested in [10].

This means that O_i is associated with two graded relations. For two labels t_i^j and t_i^k we have:
- $\Pi(t_i^j, t_i^k) = \Pi(t_i^k, t_i^j)$ assesses to what extent t_i^j and t_i^k can refer to the same thing. Note that $\Pi(t_i^j, t_i^k) = 0$ means that the two labels never refer to the same thing.

- $N(t_i^j, t_i^k)$ assesses to what extent it is certain that t_i^k is a specialization of t_i^j . N is not symmetrical. $N(t_i^j, t_i^k) = 1 = N(t_i^k, t_i^j)$ expresses that t_i^j and t_i^k are perfect synonyms. $N(t_i^j, t_i^k) = 0$ expresses a total lack of certainty that t_i^k specializes t_i^j .

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

$$N(t_i^j, t_i^h) \geq \min \left(N(t_i^j, t_i^k), N(t_i^k, t_i^h) \right), \quad (3)$$

$$\Pi(t_i^j, t_i^h) \geq N(t_i^j, t_i^k) * \Pi(t_i^k, t_i^h), \quad (4)$$

with $a * b = b$ if $b > 1 - a$ and $a * b = 0$ otherwise. (3) is the transitivity of the specialization

[15]. The “hybrid transitivity” (4) expresses that if t_i^k specializes t_i^j and if t_i^k and t_i^h can refer to the same thing, then the meanings of t_i^j and t_i^h overlaps as well (since t_i^j encompasses a larger set of situations than t_i^k); see [7] for a proof of (3)-(4) .

Moreover, we should have $\Pi(t_i^j, t_i^j) = 1$, and $\Pi(t_i^j, t_i^k) = \Pi(t_i^k, t_i^j)$. Assuming that the labels have a *clear-cut* meaning, we can state $N(t_i^j, t_i^j) = 1$. Otherwise, we only have $N(t_i^j, t_i^j) \geq \frac{1}{2}$. It is assumed that $\Pi(t_i^j, t_i^k) \geq N(t_i^j, t_i^k)$, since specialization entails that the meanings overlap. In the same way, if $N(t_i^j, t_i^k) > 0$ we should have $\Pi(t_i^j, t_i^k) = 1$, since if it is somewhat necessary (i.e. certain) that t_i^k matches t_i^j , it has to be fully possible.

So in practice, starting with a partial definition of N and Π , the relations are completed by applying repeatedly (3) and (4) and the above constraints. The other non specified values of N or Π will be assumed to be zero by default. However, remember that $\Pi(t_i^j, t_i^k) = 0$ means that the meanings of the two labels do not overlap, so the default assumption on Π is a close-world-like assumption.

In practice, binary-valued measures Π and N may be often used. We then distinguish between three cases: i) one of the labels specializes the other ($\Pi(t_i^j, t_i^k) = N(t_i^j, t_i^k) = 1$ or $\Pi(t_i^k, t_i^j) = N(t_i^k, t_i^j) = 1$); ii) the two labels have overlapping meanings ($\Pi(t_i^j, t_i^k) = 1$ and $N(t_i^j, t_i^k) = N(t_i^k, t_i^j) = 0$); iii) the meanings of the two labels are fully distinct ($\Pi(t_i^j, t_i^k) = N(t_i^j, t_i^k) = N(t_i^k, t_i^j) = 0$). This trichotomy may be refined by introducing some intermediary grades. In any case, only a small number of intermediary grades between 1 and 0 will be used. It will enable us to distinguish in particular between situations where a term is for sure a specialization of another ($N(t_i^j, t_i^k) = 1$), from situations where it is *generally* a specialization ($N(t_i^j, t_i^k) > 0$).

Requests will be also stated in terms of labels t_i^j '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_i^j 's where $t_i^j \in \mathcal{T}_i$, i.e. $T_i = \bigvee_{j \in R(T_i)} t_i^j$ where $R(T_i)$ is the set of indices involved in T_i , e.g. $T_i = \text{'hotel'}$ or 'motel' . More generally, the terms in the disjunction will be prioritized, in order to express user's preferences (e.g. 'hotel' or possibly 'motel'. See section 6).

It will be first assumed that the attribute values of data are described by means of a unique la-

bel, $T_i' = \{t_i'^k\}$ where $t_i'^k \in \mathcal{T}_i$. More generally, T_i' will not be represented by a singleton, but by a (maybe weighted) disjunction of $t_i'^k$'s for expressing that the attribute value is imprecisely known with respect to the vocabulary \mathcal{T}_i (section 7).

4. The notion of a possibilistic ontology

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

1. The lodging type, which is a label like *hotel* or *campsite*, (or more generally a disjunction)
2. The place (country, area, etc.)
3. The price, a numerical value (considered only at the end of the next section).

Possibilistic ontologies, in the sense of section 3 are used to define a vocabulary pertaining to each attribute. Considering that data are described by means of n linguistic attributes, we need to define n ontologies. However, numerical attributes (as the price in our case) do not require a restricted vocabulary, since then the information can be directly represented by a fuzzy set on the attribute domain.

Let Ω be the set of ontologies that we will use: $\Omega = \{O_i\}$ $i=1, n$. Each ontology O_i is composed of terms t_i^j : $\forall i \in \llbracket 1; n \rrbracket, O_i = \{t_i^j \in \mathcal{T}_i\}$, where \mathcal{T}_i is the vocabulary for attribute i , as defined in section 3. As already said in this section, we use possibility (Π) and necessity (N) to represent closeness in meanings, specialization and synonymy, relations in ontologies. For example, if t_i^j is an hyponym of t_i^k (i.e. t_i^j specializes t_i^k), then $N(t_i^k; t_i^j) = 1$. Otherwise, if $0 < N(t_i^k; t_i^j) \leq 1$, we are not completely sure that the term t_i^j is more specific than t_i^k . If $\Pi(t_i^k; t_i^j) \in [0, 1[$, it expresses that t_i^k and t_i^j may occasionally refer to the same thing, but that this is not generally the case (e.g. 'Great Britain' and 'sunny country').

In our example, we define two ontologies, one for the *lodging types* (Fig. 1) and one for *vacation area* (Fig. 2). These networks are simplified representations of how similarity relations between labels can be perceived. These ontologies are made up for the needs of the example, and do not claim to be genuine and complete ones. Moreover, these networks only exhibit direct linkage, since missing

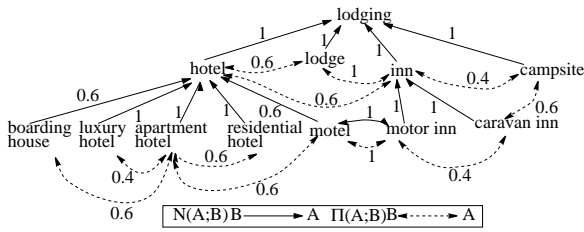


Fig. 1. Ontology defining the lodging vocabulary

weights can be recovered using (3) and (4) and the other constraints as explained in section 3.

Note that some words like *lodge* and *inn*, or *motel* and *motor inn* are only considered as *possible* synonyms. However, assessing for *lodge* and *inn*, that the possibility of referring to the same thing is 1, gives 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 considered as *inns*. Besides, when the necessity between two terms, as for *motel* and *motor inn*, is 1, the two terms are regarded 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 from the semantics of the terms. For example, $N(\text{hotel};\text{motel}) = 0.6$, means that we suppose that there exist motels that cannot be considered as hotels, but motels are generally 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 $\{0, 0.4, 0.6, 1\}$.

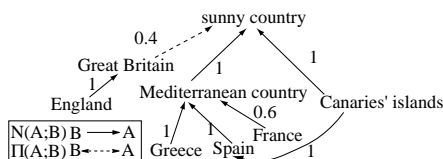


Fig. 2. Ontology defining the vacation area vocabulary

A recent work [4] considers a different form of weighted ontology, which is more oriented toward relevance issues, from an information retrieval point of view, rather than toward the evaluation of genuine similarities of meanings. The authors in [4] introduce a specialization and a generalization degree, the second degree being smaller than the first one. For instance, ‘poodle’ may be regarded as a 0.9 specialization of ‘dog’, while ‘dog’ is only a 0.4 generalization of ‘poodle’. The idea

seems to be here that a request looking for documents about ‘dog’ may be extended with a 0.9 relevance expectation to documents about ‘poodle’, while a request looking for ‘poodle’ can be extended into a request about ‘dog’ with only a 0.4 expected relevance. In our model, we also use two types of weights, but their semantics completely differ from the intended meanings of the degrees used in [4]. Indeed, the possibility weights are symmetrical on the one hand, while $N(t_i^j, t_i^k) = \alpha > 0$ does not entail anything about $N(t_i^k, t_i^j)$ even if $\alpha = 1$ on the other hand. This departs from the specialization and generalization degrees which are usually strictly positive simultaneously.

Moreover, the product is used in [4] as a transitivity operator (in place of *min* in (3) in our approach). The use of product entails a weakening of weights with the depth in the ontology. This makes the ontology “granularity dependent”. However, exploring two tree levels in specialization is better than one level in generalization (since with the above value, $0.9 \times 0.9 > 0.4$).

In our approach, the issue is more to estimate if the meaning of a label in a request covers, or just overlaps a label pertaining to the same attribute, appearing in a piece of data. Moreover, due to the use of the *min* operation, the computation of the matching degrees does not depend on the granularity of the ontology.

Building a possibilistic ontology may be a difficult task, specially if it is done by hand. Some binary ontologies exist, such as WordNet [13], exhibiting different kinds of relations between concepts. To some extent, these relations can be mapped to possibility and necessity degrees values. For instance, relations such as *synonymy* and *hyponymy* (i.e. specialization) yield necessity degrees, as already discussed. Other relations, like *meronymy* (such as *room* for *hotel*) or “see also” could be interpreted in terms of possibility degrees. A degree of possibility may reflect the category of the relation that it comes from. Another approach (e.g. [6]) could use corpus analysis and statistical co-occurrence of terms to establish relations. In any case, the ontologies should be checked and tuned by experts, even if their draft version is automatically generated. This is necessary not only because of the limitations of the automatic generation process, but also because ontologies are application-dependent and often include pragmatic information. The expert may thus weaken

some necessity links, originally graded to 1, when he knows that exceptions can be encountered (e.g. *motels* which may not be genuine *hotels*).

5. Evaluation of a request

A query is described as a set R of terms or compound terms T_i for a set of attributes. There is at most one T_i in R for each attribute i . The set R will be interpreted as a conjunction in the evaluation. Each T_i is a disjunction of terms from the vocabulary defined by the ontology associated with the attribute. Namely, $R = \bigwedge_{i \in A(R)} T_i$ and $T_i = \bigvee_{j \in R(T_i)} t_i^j$ where $A(R)$ is the set of attributes involved in the query and $R(T_i)$ is the set of the terms involved for attribute i . Each t_i^j belongs to the ontology O_i . An example of such a request is $R = (hotel \vee inn) \wedge (sunnycountry)$.

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 $\pi_i = \max_{j \in R(T_i)} \Pi(t_i^j, t_i^k)$ where $T_i' = \{t_i^k\}$. The maximum is used since T_i is defined as a disjunction. Likewise, we compute the necessity value $\nu_i = \max_{j \in R(T_i)} N(t_i^j, t_i^k)$. Note that if the data term t_i^k , is the same as the query attribute t_i^j , 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 minimum between attribute scores $\Pi(R, T') = \min_{i=1, n} \pi_i$ and $N(R, T') = \min_{i=1, n} \nu_i$. Then, the pieces 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 for $N(R, T')$.

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. 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 (compound) terms for the considered attributes, i.e. $T' = \{T_i', i \in A(R)\}$. We also assume that the data attribute values are singleton terms from the ontology, i.e. $T_i' = \{t_i^k\}$ where $t_i^k \in O_i$.

Note that if R contains a disjunction whose expression is redundant w.r.t. the ontology, i.e.

$R = t \vee t'$ and $N(t, t') = 1$ is true in the ontology, then it can be checked that the application of the two request t and $t \vee t'$ yield the same result w.r.t a database. Indeed, it is expected that enlarging a request by introducing more specialized terms in it is innocuous.

We now illustrate our approach with the following example. Let us consider the above query $R = (hotel \vee inn) \wedge (sunnycountry)$. Let us assume we have the data base:

	lodging	area	price
1	hotel	England	[65,70]
2	boarding house	Spain	25
3	lodge	Greece	cheap
4	motel	France	moderate

Let us evaluate the query. For the first row, we have $\pi_{lodging} = \max(\Pi(hotel, hotel), \Pi(inn, hotel))$ and $\pi_{area} = \Pi(sunnycountry, England)$. Obviously, $\Pi(hotel, hotel) = 1$ and according to the ontology, $\Pi(inn, hotel) = 0.6$, so $\pi_{lodging} = 1$. As *England* has an indirect relation with *sunnycountry*, we get that $\pi_{area} = 0.4$, thanks to 4. So $\Pi(R, T_1') = \min(\pi_{lodging}, \pi_{area}) = 0.4$. Moreover, we have $N(R, T_1') = 0$.

Likewise, for the second row, we have $\nu_{lodging} = \max(N(hotel, boardinghouse), N(inn, boardinghouse))$. With the ontology, we have $N(hotel, boardinghouse) = 0.6$ and $N(inn, boardinghouse) = 0$, so $\nu_{lodging} = 0.6$. In the same way, $\nu_{area} = N(sunnycountry, Spain) = 1$. Consequently, $N(R, T_2') = \min(\nu_{lodging}, \nu_{area}) = 0.6$ and $\Pi(R, T_2') = 1$. 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 $N(inn, motel)$, the ontology gives $N(inn, motorinn) = 1$ and $N(motorinn, motel) = N(motel, motorinn) = 1$. Namely, we know that a *motorinn* is a specialization of *inn* and that *motel* and *motorinn* are synonymous. We can infer that $N(inn, motel) = 1$. In the same way, we have $N(hotel, motorinn) = 0.6$ since $N(hotel, motel) = 0.6$. 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). For example, we could not infer anything about $\Pi(lodge, motorinn)$ from $\Pi(lodge, inn) = 1$ and $N(inn, motorinn) = 1$. Indeed, we know that the meanings of *lodge* and *inn* overlap and that *motorinn* specialize *inn*, but

it can correspond to a type of *inns* that are not *lodges*. As an answer, we obtain the following ranking (row, Π , N): (2, 1, 0.6); (4, 1, 0.6); (3, 1, 0); (1, 0.4, 0). Indeed row 4 is ranked after the row 2, since it has obtained the grade 0.6 for necessity on both criteria, while row 2 has a better grade on one of the query's criteria ($\nu_{area} = 1$).

Let us now consider a query component about price: $R' = (\text{moderate})$. In order to match numerical values with this linguistic term, we need to define a representation of labels such as 'moderate' in terms of price values, for a given price vocabulary, i.e. to define a (fuzzy) price distribution for each term in this vocabulary. This can be done like in figure 3. Note that ranges of prices associated with labels (e.g. *cheap*) are user- and context-dependent, and thus should be obtained from the user. This is rather easy since only endpoints of intervals have to be elicited.

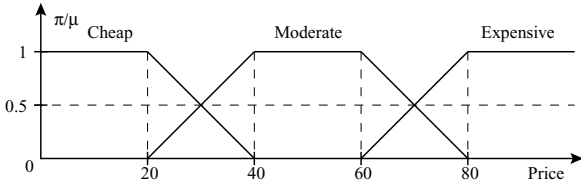


Fig. 3. Distributions of prices

Evaluating this query with classical FPM, using (1) and (2), we have the following results (row, Π , N): (4,1,0.5); (1,0.75,0.5); (3,0.5,0); (2,0.25,0); Note that the necessity $N(\text{moderate}, \text{moderate})$ is 0.5, due to the $\max(\mu_{\text{moderate}}(u), 1 - \pi_{\text{moderate}}(u))$ in (2). The evaluation on the first row for 'moderate' and 'expensive' is worth to be examined. Indeed, we have $\Pi(\text{moderate}, [65, 70]) = 0.75$ and $N(\text{moderate}, [65, 70]) = 0.5$. Observe here that the possibility is not 1 although the necessity is strictly positive, because 'moderate' has a fuzzy meaning. In the same way, $\Pi(\text{expensive}, [65, 70]) = 0.5$ and $N(\text{expensive}, [65, 70]) = 0.25$. The constraints $N(t_i^j, t_i^k) > 0 \Rightarrow \Pi(t_i^j, t_i^k) = 1$ no longer holds, since we have a fuzzy query, only the constraint of section 2, i.e. $\Pi(T; T') \geq N(T; T')$ holds. As $[65, 70]$ is nearer to 'moderate' than to 'expensive', the degrees are higher.

Such evaluations can be combined with the previous ones with *min* operation, thus allowing for the evaluation of compound queries dealing with heterogeneous data, expressed in different ways (terms, values, intervals). The combina-

tion of the matching degrees obtained by symbolic pattern matching (as described in this paper) with the matching degrees computed by fuzzy pattern matching (as recalled in section 2) raises the problem of the commensurability of the scales. Indeed, the fuzzy pattern matching applied to continuous membership functions can yield any real number in $[0,1]$, while symbolic pattern matching is supposed to use only discrete scales with a rather small number of levels, which however may be numerically encoded for convenience. Assuming that the symbolic pattern matching uses a finite scale with homogeneously distributed levels as, e.g. $\{0,0.2,0.4,0.6,0.8,1\}$, the grades obtained by fuzzy pattern matching can be approximated to their closest value on this scale. Since the two matching procedures are both based on possibility and necessity degrees, this allows us to combine the elementary evaluations computed for each attribute. For instance, if we consider the compound query $R \wedge R'$ in the above examples, we get row 4 ranked first. Then come rows 3, 1 and 2.

6. Prioritized requests

Let us now consider more general queries by allowing for *fuzzily weighted* disjunctions in them. This enables the user to express preferences in the data selection, by prioritizing the query terms.

This is 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 elementary terms are belonging to the query. We can symbolically write $T_i = \bigvee_{j \in R(T_i)} (\lambda_i^j, t_i^j)$, where λ_i^j is the weight of term t_i^j . We assume that $\lambda_i^j \in [0, 1]$ and that $\max_j \lambda_i^j = 1$, which means that there is at least one term that perfectly 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: $(0.5, \text{lodge}) \vee (0.7, \text{motel}) \vee (0.8, \text{apartment hotel}) \vee (1, \text{luxury hotel})$. The query is then evaluated by way of a weighted maximum [9]:

$$\pi_i = \max_{j \in R(T_i)} (\min(\lambda_i^j, \Pi(t_i^j, t_i^k))),$$

$$\nu_i = \max_{j \in R(T_i)} (\min(\lambda_i^j, N(t_i^j, t_i^k))).$$

Let us consider the request: $R = (\text{cosylodging}) \wedge (\text{sunnycountry})$.

For the fourth row, ν_{lodging} is no longer 0.6.

We have $\nu_{\text{lodging}} = \max(\min(0.5, 0), \min(0.7, 1))$,

$\min(0.8,0), \min(1,0) = 0.7$. Besides, $\mu_{sunnycountry} = 0.6$. The final result will be now for the triple (Row, Π, N) : $(4, 0.7, 0.6)$; $(3, 0.6, 0.5)$; $(2, 0.6, 0)$; $(1, 0.4, 0)$.

As in the price query of section 5, one of requirements in the request is fuzzy. This is why values less than 1 can be obtained for the possibility when the necessity degree is strictly positive.

The same kind of weight can be used to express a preference between the attributes themselves. In our example, the *lodging type* can be less important than the country, so that the query becomes: $R = \bigwedge_{i \in R(i)} (\omega_i, T_i)$ where ω_i has the same constraints as λ_i^j and represents the importance of the attribute in the query. Here, as we have a conjunction, the evaluation will be [9]:

$\Pi(R, T') = \min_{i \in R(i)} \max(1 - \omega_i, \pi_i)$,
 $N(R, T') = \min_{i \in R(i)} \max(1 - \omega_i, \nu_i)$. As an example, first consider the request $(0.2, (1, hotel) \vee (0.6, inn)) \wedge (1, (0.4, England) \vee (1, sunny country))$ which privileges the area attribute. This will give as a result: $(2, 1, 0.8)$; $(3, 0.8, 0.8)$; $(4, 1, 0.6)$; $(1, 0.4, 0.4)$. Take now the request $(1, (1, hotel) \vee (0.6, inn)) \wedge (0.7, (0.4, England) \vee (1, sunny country))$ where the lodging is more important; we get: $(4, 1, 0.6)$; $(2, 1, 0.6)$; $(1, 0.4, 0.4)$; $(3, 0.6, 0)$. Lowering the importance of the area in the query leads to prefer row 4, which corresponds to a place which is not sunny with full certainty (nor England). Row 1 is now preferred to row 3. Indeed, the lodging type is better for row 1 than for row 3, although Greece is sunnier than England. It is why giving more importance to the lodging type than to the area improves the ranking of the row 1.

7. 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 in the previous sections, can now be a set, or more generally a fuzzy set of terms. We now have $T'_i = \bigvee_{k \in D(T'_i)} \lambda_i^k / t_i^k$, 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:

$\Pi(T_i, T'_i) = \max_{j,k} \min(\lambda_i^j, \lambda_i^k, \Pi(t_i^j, t_i^k))$
 $N(T_i, T'_i) = \max_{j \in R(T_i)} \min(\lambda_i^j, \min_{k \in D(T'_i)} n_i^k)$
 where $n_i^k = \max(1 - \lambda_i^k, N(t_i^j, t_i^k))$. The formula

giving $N(T_i, T'_i)$ expresses that it should exist a term t_i^j in the request such that all the t_i^k 's appearing in T'_i are specializations of t_i^j . Indeed, the description of the attribute value of a 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_i^k is a specialization of t_i^j is all the less compulsory as λ_i^k is small (at the extreme, when $\lambda_i^k = 0$, i.e. t_i^k does not appear in T'_i , $N(t_i^j, t_i^k)$ should have no influence).

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

$\pi_{lodging} = \max(\min(1, 1, 1), \min(1, 1, 0.5)) = 1$
 $\nu_{lodging} = \min(1, a) = 0.6$ with
 $a = \min(\max(1 - 1, 1), \max(1 - 0.5, 0.6)) = 0.6$
 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 $\nu_{lodging} < 1$.

This result would remain the same if we had : $D = \{ (1, hotel) \vee (1, motel); France \}$, since the max operation keeps the necessity degree $N(t_i^j; t_i^k)$ and not the value of the weight. To change the ranking, the importance of the term (here *motel*) has to be decreased such that $\lambda_i^k < 1 - N(t_i^j; t_i^k)$, which defines an importance threshold. Indeed, with $D = \{ (1, hotel) \vee (0.3, motel); France \}$, we would have $\nu_{lodging} = 0.7$. This can be compared with $D = \{ (0.5, hotel) \vee (1, motel); France \}$, giving also $\pi_{lodging} = 1$ and $\nu_{lodging} = 0.6$. In this latter case, changing the *hotel* weight does not change the degrees, since $N(hotel, motel) = 0.6$. Besides, it can be checked that if the weight of *motel* is set to 0, we recover $\nu_{lodging} = 1$, as the query and the data match perfectly (which is indeed equivalent to having *hotel* only in the description of the piece of data).

Note that if the data attribute is a disjunction of perfect synonyms $(t_i^1 \vee t_i^2)$ having the same weight, the evaluation will be equivalent to considering only one of the terms. Indeed $\Pi(t_i^1, t_i^2) = 1$ and $N(t_i^1, t_i^2) = N(t_i^2, t_i^1) = 1$, since the two terms are perfectly matching, and thus it can be checked that $N(t_i^j, t_i^1) = N(t_i^j, t_i^2)$, using (3) and $\Pi(t_i^j, t_i^1) = \Pi(t_i^j, t_i^2)$ due to (4).

8. Textual data

The approach presented above can be adapted with some modifications to textual information. A piece of text is viewed as a list of single words where non-significant words are dropped. However, we focus our attention on short texts, such as e.g. titles, where information retrieval approaches based on statistics on the occurrence of words do not apply. A row in the database, as in the above approach, is now replaced by a conjunction of significant words. The evaluation of a query, represented as a weighted disjunction or conjunction of keywords, can then be done as in the spirit of the previous sections.

However, some new issues are raised when dealing with this kind of data. As the ontology refers to concepts, which can be labeled by groups of words, these concepts should be recognized in the text. This remark shows the limitation of the use of ontologies in this kind of search, when natural language processing techniques are not used (as it is often the case in information retrieval). Indeed, the natural method would be to use a tagger and a parser in order to identify the conceptual categories of the words and concepts, as done in [4] for example. To really deal with texts, we would need at least a simple textual analyzer, making a parsing or an indexing phase, but this is not the issue considered in this section.

An easier (but more naïve) solution might be to identify phrases by grouping words according to proximity and match them with the ontology. If the ontology only contains single words or phrases that directly appear in the text (up to a lemmatization of the words), our pattern matching procedure can be applied.

Let's consider the following database containing titles of papers:

1	Dealing with vagueness of natural languages
2	Tolerant fuzzy pattern matching: an introduction
3	A hierarchical model of fuzzy classes
4	Resolution principles in possibilistic logic
5	Weighted fuzzy pattern matching
6	Flexible queries to a crisp database

In fact, we take into account the neighbourhood of each term in the evaluation, in order to solve the problem illustrated by title 6. Namely, suppose we are looking for articles about ‘fuzzy databases’.

Then, even if we *know* (from the ontology) that ‘flexible’ and ‘fuzzy’ are often synonyms ($N=0.8$), and therefore if we replace the former by the latter term in the sixth paper title, the title should not match the query since what is sort of ‘fuzzy’ in this article is the ‘queries’ but not the ‘database’, which is in fact ‘crisp’! To deal with such a case, we need to take into account the proximity of the terms in the text.

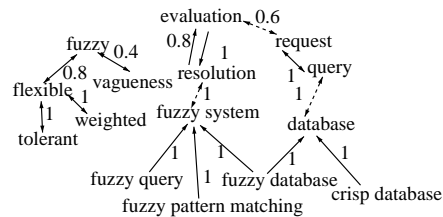


Fig. 4. Ontology fragment for the title example

Therefore, there are two aspects of word proximity that we should manage: the phrases identification and the word context for the query's scope. As it has been mentioned, we do not want to deal with natural language processing aspects here, but we just provide a further illustration of the potentials of our approach.

Let us consider the example made by the above title database and the ontology exhibited on figure 4 that shows a fragment of a domain-oriented ontology dealing with topics or keywords of articles.

Let us first evaluate the simple query: $R = \text{fuzzy}$. We have $N(\text{fuzzy}, \text{vagueness}) = 0.4$, so $N(R, D_1) = 0.4$. Obviously, $N(R, D_3) = 1$ and $N(R, D_4) = 0$. As $N(\text{fuzzy}, \text{flexible}) = 0.8$, paper 6 is also retrieved with $N(R, D_6) = 0.8$. In the same way, $N(\text{fuzzy}, \text{tolerant}) = 0.8$, as ‘tolerant’ and ‘flexible’ are almost synonyms. Therefore, we have $N(R, D_2) = \max(0.8, 1) = 1$, since D_2 contains both terms (idem for D_5). A slightly more complex query looking for papers dealing with ‘fuzzy request’ is understood as $\text{fuzzy} \wedge \text{request}$ because the expression *fuzzy request* does not exist as such in the ontology, even when considering transitivity properties. Since $N(\text{request}, \text{query}) = 1$, D_6 is the only paper which can be judged as relevant with $N(R, D_6) = \min(0.8, 1) = 0.8$. Let us now consider the query ‘fuzzy database’. The label *fuzzy database* exists in the ontology, so the query is interpreted as $R = \text{fuzzy_database}$ and not $\text{fuzzy} \wedge \text{database}$. In the same way, by grouping words in the title D_6 , the ontology label *crisp*

database is also recognized. The matching is therefore done between these two labels. This leads to $N(\text{fuzzy_database}, \text{crisp_database}) = 0$ and $\Pi(\text{fuzzy_database}, \text{crisp_database}) = 0$, therefore no results are retrieved. Whereas if the words had not been grouped together, the evaluation with only *database* would have given D_6 as a result, since $\Pi(\text{fuzzy_database}, \text{database}) = 1$, which is clearly undesirable.

This short example shows that we can take advantage of the ontology itself for handling the word context problem mentioned above.

9. 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^j and t_i^k are reflected by means of two indexes. $\Pi(t_i^j, t_i^k)$ assesses to what extent t_i^j and t_i^k can refer to the same thing, while $N(t_i^j, t_i^k)$ evaluates to what extent it is certain that t_i^k specializes t_i^j . 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 jointly handle attributes valued by linguistic terms with numerical attributes.

A prototype has been implemented to validate the examples given in this paper. The only noticeable difficulty in the implementation is the problem of efficiently propagating the constraints (such as (3) and (4)), for evaluating the possibility and necessity degrees between two terms. In case of large ontologies, it may be done preferably offline.

An extension of the qualitative pattern matching technique to textual data has been also presented here, and has been started to be explored in a multilingual context [3].

References

- [1] T. Andreasen, H. Christiansen, and H. L. Larsen, editors. *Flexible Query Answering Systems*. Kluwer, 1997.
- [2] A. Bidault, C. Froidevaux, and B. Safar. Similarity between queries in a mediator. In *Proc. 15th European Conference on Artificial Intelligence*, pages 235–239. ECAI'02, Lyon, July 2002.
- [3] M. Boughanem, Y. Loiseau, and H. Prade. Graded pattern matching in a multilingual context. In B. De Baets, J. Fodor, and G. Pasi, editors, *Proc. 7th Meeting Euro Working Group on Fuzzy Sets*, pages 121–126. Eurofuse, Varena, September 2002.
- [4] H. Bulskov, R. Knappe, and T. Andreasen. On measuring similarity for conceptual querying. In *Flexible Query Answering Systems, LNAI 2522*, pages 100–111. Springer, 2002.
- [5] M. Cayrol, H. Farreny, and H. Prade. Fuzzy pattern matching. *Kybernetes*, 11:103–116, 1982.
- [6] C.J. Crouch. An approach to the automatic construction of global thesauri. *Information Processing and Management*, 26(5):629–640, 1990.
- [7] D. Dubois and H. Prade. Resolution principles in possibilistic logic. *Int. Jour. of Approximate Reasoning*, 4(1):1–21, 1990.
- [8] 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.
- [9] D. Dubois, H. Prade, and C. Testemale. Weighted fuzzy pattern matching. *Fuzzy Sets and Systems*, 28:313–331, 1988.
- [10] 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.
- [11] D. Kraft, G. Bordogna, and G. Pasi. Fuzzy set techniques in information retrieval. In J. Bezdek et al., editor, *Fuzzy Sets in Approximate Reasoning and Information Systems*, chapter 8, pages 469–510. Kluwer, 1999.
- [12] Y. Loiseau and H. Prade. Qualitative pattern matching with linguistic terms. In T. Vidal and P. Liberatore, editors, *STAIRS 2002*, pages 125–134. STArting AI Researchers Symp., Lyon, IOS Press, July 2002.
- [13] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244, 1990.
- [14] P. Resnik. Semantic similarity in a taxonomy: an information-based measure and its application to problem of ambiguity in natural language. *J. Artif. Intellig. Res.*, 11:95–130, 1999.
- [15] 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.
- [16] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Trans. on Fuzzy Systems*, 4(2):103–111, 1996.