

Precision-Controlled Retrieval of Qualitative Information from Databases Using Hierarchies

Adolfo Guzman-Arenas^{1,2}, Serguei Levachkine¹, and Victor-Polo de Gyves²

¹Centre for Computing Research (CIC) - National Polytechnic Institute (IPN)

UPALMZ, CIC Building, 07738, Mexico City, MEXICO

²SoftwarePro International

a.guzman@acm.org sergei@cic.ipn.mx degyves@gmail.com

Abstract. Can we retrieve objects from a database that are just close to a desired item? Can we control the retrieval process up to a given error? Can we define a semantic similarity between objects to be retrieved? The paper seeks for positive answers to these questions by using hierarchies. Sometimes, qualitative values can be arranged in a *hierarchy*, a tree with certain properties. Hierarchies allow the definitions of following constructs: the confusion $\text{conf}(r, s)$ in using qualitative value r instead of the intended or correct value s (such as using ‘umbrella’ when ‘raincoat’ was intended); a predicate P that holds for object o within confusion ε , written $P_\varepsilon(o)$; the *closeness* of an object to a predicate (closeness of fit); how close or similar two objects are; predicate $P^\varepsilon(o)$, where the total (cumulative) error produced by object o is at most ε . These constructs are used in an extended SQL notation to retrieve, from a database, answers imperfectly fulfilling a predicate, up to a desired confusion (error). The extended SQL expression (a query) is automatically transformed to normal SQL, which then retrieves. It is shown how to extend *any* relational database for precision-controlled retrieval and updates. Examples are given.

Keywords: Information Retrieval, Hierarchy, Semantic Similarity, Extended SQL, confusion

1 Introduction

We are observing day-by-day more and more simple procedures of data content acquisition, processing, and transmitting. However, these constitute only one part of the problem. The other part is that access to heterogeneous and independent databases should be equally simple. Unfortunately, identification of desired (useful) information by searching and filtering has become more and more complicated [1]. That is why the treatment of differences in the structure and semantics of the data stored in repositories plays an increasingly important role in modern information systems [1], [2]. The first studies on interoperating information systems have been directed toward syntactic (i.e., data types and formats) and structural heterogeneities (i.e., schematic integration, query languages, and interfaces) [1], [3]. As interoperating information systems increasingly confront more complex knowledge management tasks the technology needed to deal successfully with these issues must focus on the semantics underlying the data used by those systems [1], [4].

Recent investigations in information retrieval and data integration have emphasized the use of ontologies and semantic similarity functions as a mechanism for comparing objects that can be retrieved or integrated across heterogeneous repositories [5], [6], [7], [8], [9]. In this context, ontology is a type of knowledge base that describes concepts through definitions that are sufficiently detailed to capture the semantics of a domain. Ontology represents a certain view of the world, supports intentional queries regarding the content of database, and reflects the relevance of data by providing a declarative description of semantic information independent on the data representation [10].

Our current work is motivated by the need of new tools that can improve the retrieval and integration of information. In present paper, we focus on hierarchies – a simpler, albeit very useful, version of ontologies. They are easier to understand, to implement, and their application to the extensions to searches, queries, and imperfect answers are straightforward [11]. Ontologies promise longer mileage, although they are more complex to understand, to implement, and to apply. For instance, *BiblioDigital* is a recent development that uses

for document classification and indexing a rich taxonomy, like ontology, but with the similarity properties, like a hierarchy [12].

A datum makes sense only within a context. Intuitively, we know that “computer” is closer to “office” than to “ocean” or to “dog.” A “cat” is closer to “dog” than to “bus station.” “Burning” is closer to “hot” than to “icy.” How can we measure these similarities? What wearing apparel do we wear for rainy days? *Raincoat* is a correct answer; *umbrella* is a close miss; *belt* a fair error, and *typewriter* a gross error. What is closer to an *apple*, a *pear* or a *caterpillar*? Can we measure these errors? How related or close are these words?

Our approach to modeling similarity between items from the same hierarchy or from different hierarchies is based on an asymmetric, context-dependent measure called *confusion* (in using a qualitative value instead of the intended or correct value). The latter term is introduced to differentiate it from traditional approaches that used different kind of *distances* (i.e., symmetric, context-independent measures) for this purpose. The confusion’s asymmetry is given by definition and its context dependence by hierarchical structure. The concept of confusion allows defining the closeness to which an object fulfills a predicate as well as deriving other operations and properties among hierarchical values.

The remainder of this paper is structured as follows: Section 2 reviews current approaches to comparing objects and concepts across ontologies. Different types of hierarchies are defined in Section 3. Section 4 describes properties and functions on hierarchies: the confusion in using a qualitative value instead of the intended or correct value; a predicate that holds for an object within a given confusion; the *closeness* of an object to a predicate; how close or similar two objects are; a predicate, where the total (cumulative) error produced by an object is at most a given confusion. In Section 5, the constructs of Section 4 are used in an extended SQL notation to retrieve from a database answers imperfectly fulfilling a predicate up to a desired confusion (error). The extended SQL expression (a query) is automatically transformed to normal SQL, which then retrieves. It is shown how to extend *any* relational database for precision-controlled retrieval and updates. Section 6 presents our conclusions.

2 Previous Work

In environments with multiple information systems, independent systems may have their own intended models and, therefore, their own ontologies [13]. In such environments, the general approach to data integration has been to map the local terms of distinct ontologies onto a single shared ontology. Then, the semantic similarity is typically determined as a function of the path distance between terms in the hierarchical structure underlying the single ontology [14], [15], [16], [17]. Other methods to assess semantic similarity within a single ontology are feature matching [18] and use of information content [7], [19]. The feature-matching approach uses common and different characteristics between objects or entities to compute semantic similarity. Information content, on the other hand, uses information theory [20] to define similarity measure in terms of the degree of informativeness of the immediate superconcept that subsumes the two concepts being compared.

The use of a single ontology does ensure complete integration across heterogeneous information systems. However, this type of ontology is costly, if not impractical to obtain, since users and information systems are forced to commit to this single ontology and compromises are difficult to maintain when new concepts are considered. Using another approach, which considers scalability issues in building an ontology, some studies create a shared ontology by integrating existing ones [21], [22], [23], [24]. Ontology integrations need to treat overlapping concepts and inconsistencies across ontologies. Like semantic heterogeneity in the database field [25], ontology mismatches occur when two ontologies have terms denoting categories, components of category definitions, or ontological concepts that are the same [26].

A strategy for ontology integration is the mapping of local ontologies onto a more generic ontology [13], [24], [27]. For example, ONIONS [24] is a methodology for ontology analysis and integration that has been applied to large medical terminologies. Ontology integration in ONIONS is done by formally representing all concepts and by ontologically integrating these concepts through a set of generic ontologies. The use of semantic interrelations is another approach for ontology integration. For example, OBSERVER is an ontology-

based system that is enhanced with relationships for vocabulary heterogeneity resolution [21], [22], [28]. It uses terminological relations (hyponymy and hypernymy) to map the nontranslated terms in a user ontology onto terms (which are not synonymous) in a target ontology. This translation process is recursive and consists of substituting nontranslated terms with the intersection of their immediate parents or the union of their immediate children. Similarly, [23] used synonymy and hyponymy terminological relations for ontological integration, but it also included a relation of positive association that connects terms generally used in the same context. This approach is semiautomatic and starts with the extraction of hyponyms and associated terms from the source schema. Synonyms and domain-related knowledge are introduced by a person responsible for the integration. A validation of terminological relations defined for attributes in the ontology is then followed by the inference of new relations.

Once ontologies have been integrated, similarity measures are applied to compare concepts in much the same way as a similarity evaluation is done within a single ontology. A recent work presents different measures for comparing concepts whose formal definitions support inferences of subsumption and whose local concepts in differentiated ontologies inherit their definitional structures from concepts in a shared ontology [27]. This study assumes that the set intersection of concepts' instances is an indication of these concepts' correspondence. Three main types of measures for comparing concepts descriptions are discussed: 1) filter measures based on a path distance, 2) matching measures based on graph matching that make one-to-one correspondence between elements of concepts' descriptions, and 3) probabilistic measures that give the correspondence in terms of the joint distribution of concepts.

In general, current methods that compare concepts from different ontologies are based on an a priori integration of local ontologies through a top-level ontology or through terminological relations that are defined manually or semiautomatically. An interesting exception is [29]. This study presents a computational approach that compares concepts from unconnected and independent ontologies without constructing a priori a shared ontology. Its approach to modeling similarity is based on a matching process [18] that uses available

information from various ontology specifications (i.e., synonym sets, distinguishing features, and semantic relations of entity classes). Such similarity modeling establishes links among ontologies while keeping each ontology autonomous. This is a weak form of integration because it does not allow deep processes, that is, it cannot be used for making inferences about the relationship among other entity classes within a given ontology and cannot guarantee computations that require particular components of the entity class representation. It provides, however, a systematic way to detect which entity classes are most similar to each other and, therefore, which entity classes are the best candidates for establishing integration across the ontologies. Our present work shares similar ideas.

We are closing this section with a brief overview of other related works in the fields of Artificial Intelligence, Databases, Natural Language, Knowledge Representation, and so on. Their communities have been gauging the distance, proximity or “relatedness” between symbolic values.

Hierarchies. The concept of a (generalization) hierarchy is not new. Hierarchies are used in data warehousing and data mining; see, for instance, the H-sets in [30]. A practical use of hierarchies in symbolic processing is Clasitex [31], which finds the themes of an article written in Spanish or English. It uses the concept tree, and a word (not in the tree) *suggests the topic of* one or more concepts in the tree. *BiblioDigital*© [12], a recent development, uses a large taxonomy (although not a hierarchy) to classify text documents; a (distributed) crawler in it retrieves “external” documents residing elsewhere in the Web. If a document is about (Cf. Clasitex) war, Iraq and President Bush, its URL will be stored in these three nodes in the concept tree. Hierarchies are simpler than ontologies, albeit very useful [32].

The data modeling community, through the entity-relationship model, also organize items by their nature, properties, and the relations among them.

Natural Language. Linguists have proposed many versions of semantic closeness, similarity, and other measures among words. For example, [33] identifies conceptually similar *documents* using a single ontology. [34] does the same using a topic hierarchy - a kind of ontology. [35] builds trees of words, and by graph

matching retrieves similar texts. Another common idea twisting around is to regard the representation space with a “universal” measure of proximity of space’s elements and then an attempt to adapt it to different subject domains [36], [37].

WordNet (A Lexical Database for the English Language [http:// www.cogsci.princeton.edu/~wn/](http://www.cogsci.princeton.edu/~wn/)) organizes information in logical groupings called synsets; each synset is a list of synonymous words or collocations (e.g., “fountain pen”, “take in”), and pointers that describe the relations between this synset and other synsets. A word or collocation may appear in more than one synset, and in more than one part of speech. The words in a synset are logically grouped such that they are interchangeable in some *context*. Nouns and verbs are organized into *hierarchies* based on the hypernymy/hyponymy relation between synsets. Two kinds of relations are represented by pointers: lexical and semantic. Lexical relations hold between word forms; semantic relations hold between word meanings. These relations include (but are not limited to), antonymy, entailment, and meronymy/holonymy. Additional pointers are used to indicate other relations.

[38] compares five measures of similarity or semantic distance in WordNet: Jiang and Conrath's measure (the best in the comparison: a spelling-corrector on real data [7]); that of Hirst-St-Onge (seriously over-related), that of Resnik (seriously under-related [37]), and those of Lin [36], and of Leacock-Chodorow (in between). Note that all the measures except those of Hirst and St-Onge are *similarity* (not relatedness) measures considering only *the hyponymy hierarchy* of WordNet. The main problem with these approaches is that they use *distances*, thus obeying the symmetric property, while *conf* does not.

Ontologies. At least three approaches appear when measuring similarity or relatedness of concepts (nodes in the ontology):

1. *Syntactic approach.* Methods that take into account only the organization of the tree or data structure of the ontology; for instance [35], those based on XML, or the “ontology merging” of Protégè [39].
2. *Standard ontology.* Use of a common or agreed-upon ontology. Clearly, if different people (or agents) use the *same* ontology, similarities among concepts will be consistently measured across us-

ers. CYC [40] was an early attempt to build the concept tree for common concepts. A common ontology is predicted in [41]; conceptually similar *documents* are identified in [33] by using a single ontology. In contrast, point (3) following shows use of different ontologies.

3. *Measuring similarity across ontologies.* LIA, a language for agent interaction [32], [42] has an ontology comparator COM that maps a concept from one ontology into the closest corresponding concept in another ontology by computing *sim* function. By repeated use of *sim*, the *degree of understanding* $du(B, O_A)$ of agent B (with ontology O_B) about ontology O_A is found in [43].

Instead of using ontologies, this paper works on arbitrary *hierarchies*. Why? Because the problem-oriented interaction can be easier to maintain if the hierarchical structure is not a priori rigid as in the case of common hierarchies or ontologies.

Pattern Classifiers. Our predicates with controlled precision or confusion are similar to Pattern Classifiers [44], but these classify *objects* according to the values of their properties, whereas hierarchies help to classify these *values*, when they are non-numeric.

Distances and ultradistances. Traditionally [45], the representation space is regarded as a metric space with some “exotic” distance (e.g., ultrametric distance to measure the “distances” between members of a hierarchy). However, often is not the case that such a distance meets the needs of the classification problem under consideration. Thus, we lean towards functions like *conf* that are not distances.

3 Hierarchies of Qualitative Values and Qualitative Variables

In this section we introduce the concepts of hierarchy, qualitative variable, symbolic value, hierarchical variable, as well as the types of hierarchies to be considered along this paper.

3.1 The concept of hierarchy

Definition 3.1 (Element set). A set E whose elements are explicitly defined. ♦¹

Example 3.1: {red, blue, white, black, pale}.

Definition 3.2 (Ordered set). An element set whose values are ordered by a $<$ (“less than”) relation. ♦

Example 3.2: {very_cold, cold, warm, hot, very_hot}.

Definition 3.3 (Covering). K is a covering for set E if K is a set of subsets $e_i \subset E$, such that $\cup_i e_i = E$. ♦

Every element of E is in some subset $e_i \in K$. If K is not a covering of E , we can make it so by adding a new e_j to it, named “others”, that contains all other elements of E that do not belong to any of the previous e_i .

Definition 3.4 (Exclusive set). K is an exclusive set if $e_i \cap e_j = \emptyset$, for every $e_i, e_j \in K$. ♦

Its elements are mutually exclusive. If K is not an exclusive set, we can make it so by replacing every two overlapping $e_i, e_j \in K$ with three: $e_i - e_j$, $e_j - e_i$, and $e_i \cap e_j$.

Definition 3.5 (Partition). P is a partition of set E if it is both a covering for E and an exclusive set.

Definition 3.6 (Qualitative variable). A single-valued variable that takes symbolic values. ♦

Its value cannot be a set². By symbolic we mean qualitative, as opposed to numeric, vector or quantitative variables.

¹ The symbol ♦ means: end of definition.

² Variable, attribute and property are used interchangeably. An object may have an attribute (*Example:* weight) while others do not: the weight of blue *does not make sense*, as opposed to saying that the weight of blue *is unknown* or not given. A variable (*color, height*) describes an aspect of an object; its value (*blue, 2 Kg*) is such description or measurement.

Definition 3.7 (Symbolic value). A symbolic value v represents a set E , written $v \propto E$, if v can be considered a name or a depiction of E . ♦

Example 3.3: pale \propto {white, yellow, orange, beige}.

Definition 3.8 (Hierarchy). For an element set E , a hierarchy H of E is another element set where each element e_i is a symbolic value that represents either a single element of E or a partition, and $\cup_i \{ r_i \mid e_i \propto r_i \} = E$ (The union of all sets represented by the e_i is E). ♦

Example 3.4 (Hierarchy H₁): for $E = \{\text{Canada, USA, Mexico, Cuba, Puerto_Rico, Jamaica, Guatemala, Honduras, Costa_Rica}\}$, a hierarchy H_1 is $\{\text{North_America, Caribbean_Island, Central_America}\}$, where North_America \propto {Canada, USA, Mexico}; Caribbean_Island \propto {Spanish_Speaking_Island, English_Speaking_Island}; Spanish_Speaking_Island \propto {Cuba, Puerto_Rico}; English_Speaking_Island \propto {Jamaica}; Central_America \propto {Guatemala, Honduras, Costa_Rica}³.

Hierarchies make it easier to compare qualitative values belonging to it (§4).

Definition 3.9 (Hierarchical variable). A hierarchical variable is a qualitative variable whose values belong to a hierarchy (The data type of a hierarchical variable is hierarchy). ♦

Example 3.5: place_of_origin that takes values from H_1 .

Note 3.1: Hierarchical variables are single-valued. Thus, a value for place_of_origin can be North_America or Mexico, but not {Canada, USA, Mexico}, although North_America \propto {Canada, USA, Mexico}. When a tree represents a hierarchy, the partition of each node is shown as descendants (subsets) of that node.

Note 3.2: We will also write a hierarchy such as H_1 thus: {North_America \propto {Canada USA Mexico} Caribbean_Island \propto {Spanish_Speaking_Island \propto {Cuba Puerto_Rico} English_Speaking_Island \propto {Jamaica} } Central_America \propto {Guatemala Honduras Costa_Rica} }, sometimes omitting the symbol \propto for simplicity.

³ An element *other* is added by default, where it is needed to complete the hierarchy partitions.

Definition 3.10. We are also going to use the following common notations: a) **father_of** (v). In a tree representing a hierarchy, the father_of a node is the node from which it hangs; b) the **sons_of** (v) are the values hanging from v . The nodes with the same father are **siblings**; c) **grand_father_of**, **brothers_of**, **aunt**, **ascendants**, **descendants**... are defined, when they exist; d) The **root** is the node that has no father. ♦

3.2 Simple, ordered, percentage, and mixed hierarchies

A hierarchy describes the structure of qualitative values in a set E . We define the following hierarchies:

Definition 3.11 (Simple hierarchy). A simple (normal) hierarchy is a tree with root E and if a node has children, these form a partition of the father. ♦

A simple hierarchy describes a hierarchy where E is a set (thus its elements are not repeated, not ordered).

Example 3.6: live being {animal {mammal, fish, reptile, other animal}, plant {tree, other plant}}.

Definition 3.12 (Ordered hierarchy). In an ordered hierarchy, the nodes of some partitions obey an ordering relation. ♦

Example 3.7: object {tiny, small, medium, large}*⁴.

Definition 3.13 (Percentage hierarchy). In a percentage hierarchy, the size of each set is known. ♦

Example 3.8: AmericanContinent(740M) {North America(430M) {USA(300M), Canada(30M), Mexico(100M)} Central America (10M), South America(300M)}⁵.

Definition 3.14 (Mixed hierarchy). A mixed hierarchy combines the three former types. ♦

For these four types of hierarchies we define $\text{conf}(r, s)$ as the confusion or error in using value r instead of s , the intended or correct value (see, for instance Table 1, where $\text{conf}(r, s)$ for the elements of the hierarchy H_2 of Figure 1 is shown). These definitions agree with the human sense of estimation in closeness for several wrong but approximate answers to a given question; each is applicable to particular endeavors.

⁴ Notation: an * is placed at the end of the partition to signify that it is an *ordered* partition.

⁵ The size of each set is written in parenthesis after the set. Here is the number of inhabitants.

Then, we define an enriched SQL syntax that deals with approximate queries on elements in a database holding qualitative values hierarchically structured. This enriched SQL uses precision-controlled predicates. Finally, we explain how the extension (to precision-controlled retrieval) of *any* database is possible.

4 Properties and Functions on Hierarchies

I ask for a *European car*, and I get a *German car*. Is there an error? Now, I ask for a *German car*, and a *European car* comes. Can we measure this error? Can we systematize or organize these values? Hierarchies of symbolic values allow measuring the similarity between these values, and the error when one is used instead of another.

4.1 Confusion in using r instead of s for simple hierarchies

Definition 4.1. If $r, s \in H$, then the confusion in using r instead of s , written $\text{conf}(r, s)$, is:

- $\text{conf}(r, r) = \text{conf}(r, s) = 0$, where s is any ascendant of r ; (1)
- $\text{conf}(r, s) = 1 + \text{conf}(r, \text{father_of}(s))$. \blacklozenge (2)

To measure conf , count the *descending* links from r to s , the replaced value. To differentiate from other known distances, symmetric measures, linguistic terms like relatedness, closeness, similarity or semantic distance, we introduce a new term: **confusion**⁶.

Example 4.1 (Hierarchy H₂): $\text{conf}(r, s)$ for H_2 of Figure 1 is given in Table 1.

The confusion thus introduced *resembles reality* and *catches the hierarchy semantics*. For example, $\text{conf}(\text{animal}, \text{live_being}) = 0$: if they ask you for a live being and you give them an animal, the error of using animal instead of live being is 0, since all animals are live beings. Giving a live being when asked for an ani-

⁶ The functions conf herein and hereafter are not *distances*, nor *ultra-distances* [45], nor are they symmetric. However, confusions obey the triangle law.

mal has error 1; $\text{conf}(\text{live_being}, \text{animal}) = 1$. The confusion among two brothers (say, dog and cat) is 1; using a son instead of the father produces $\text{conf}=0$; using the father instead of the son makes $\text{conf} = 1$.

4.1.1 Confusion in using r instead of s , for ordered hierarchies

Definition 4.2. For hierarchies formed by sets that are lists (ordered sets), the confusion in using r instead of s , $\text{conf}^\circ(r, s)$, is defined as:

- $\text{conf}^\circ(r, r) = \text{conf}^\circ(r, \text{any ascendant of } r) = 0$;
- If r and s are distinct brothers, $\text{conf}^\circ(r, s) = 1$ if the father is not an ordered set; else, $\text{conf}^\circ(r, s) =$ the relative distance from r to $s =$ the number of steps needed to jump from r to s in the ordering, divided by the cardinality-1 of the father; (3)
- $\text{conf}^\circ(r, s) = 1 + \text{conf}^\circ(r, \text{father_of}(s))$. \blacklozenge

This is like conf for *hierarchies formed by sets*, except that there the error between two brothers is 1, and here it is a number ≤ 1 .

Example 4.2: $\text{Temp} = \{\text{icy}, \text{cold}, \text{normal}, \text{warm}, \text{hot}, \text{burning}\}$; in this list, $\text{conf}^\circ(\text{icy}, \text{cold}) = 1/5$, while $\text{conf}^\circ(\text{icy}, \text{burning}) = 5/5 = 1$.

4.1.2 Confusion in using r instead of s , for percentage hierarchies

Now consider a hierarchy H (of an element set E) but composed of bags (unordered collection where repetitions are allowed) instead of sets.

Definition 4.3. For bags, the confusion in using r instead of s , $\text{conf}^{\circ\circ}(r, s)$, is:

- $\text{conf}^{\circ\circ}(r, r) = \text{conf}^{\circ\circ}(r, s) = 0$, when s is any ascendant of r ;
- $\text{conf}^{\circ\circ}(r, s) = 1 - \text{relative proportion of } s \text{ in } r$. \blacklozenge^7 (4)

⁷ Number of elements of E that are in r and that also occur in s / number of elements of E that are also in r = relative popularity or percentage or proportion of s in r . Note that this definition takes into account the information content.

Example 4.3: If $\text{baseball_player} \propto \{\text{pitcher catcher base_player} \propto \{\text{baseman baseman baseman}\}$
 $\text{field_player} \propto \{\text{fielder fielder fielder}\}$ shortstop}; then (a) $\text{conf}''(\text{fielder}, \text{baseball_player}) = 0$; (b) $\text{conf}''(\text{baseball_player}, \text{fielder}) = 1 - 1/3 = 2/3$; (c) $\text{conf}''(\text{baseball_player}, \text{left_fielder}) = 8/9$ (a left_fielder is one of those three fielders); (d) $\text{conf}''(\text{base_player}, \text{fielder}) = 2/3$.

4.1.3 Confusion in using r instead of s , for mixed hierarchies

Definition 4.4. To compute $\text{conf}''(r, s)$ in a mixed hierarchy, a simple-minded way⁸ is:

- apply rule (1) to the *ascending* path from r to s ;
- in the descending path, use rule (3) instead of rule (2), if p is an ordered set⁹; or use rule (4) instead of (2), when sizes of p and q are known. ♦

That is, use (4) instead of (2) for percentage hierarchies. This definition is consistent with and reduces to previous definitions for simple, ordered, percentage, and mixed hierarchies. The paper will derive results for conf ; those for conf' , conf'' , and conf''' can be similarly derived.

4.2 The set of values that is equal to another up to a given confusion

Definition 4.5. A value u is equal to value v , within a given confusion ε , written $u =_{\varepsilon} v$, iff $\text{conf}(u, v) \leq \varepsilon$ (It means that value u can be used instead of v , within error ε)¹⁰. ♦

Example 4.4: If $v = \text{lemon}$ (Figure 1), then (a) the set of values equal to v with confusion 0 is $\{\text{lemon}\}$; (b) the set of values equal to v with confusion 1 is $\{\text{citric lemon}\}$; (c) the set of values equal to v with confusion 2

⁸ In certain sense, there is no “proper” way to mix apples and oranges.

⁹ Here, p and q are two consecutive elements in the path from r to s , where q immediately follows p . That is, $r \rightarrow \dots \rightarrow p \rightarrow q \rightarrow \dots \rightarrow s$.

¹⁰ Notice that $=_{\varepsilon}$ is neither *symmetric* nor *transitive*.

is {plant citric pine lemon}.

4.2.1 Queries

Objects possessing several properties (or variables), some of them perhaps hierarchical variables, can best be stored as rows of a table in a relational database. We now extend the notion of queries to tables with hierarchical variables¹¹, by defining the set of objects that satisfy predicate P within a given confusion ε .

Definition 4.6. P holds for object o with confusion ε , or P holds for o within ε ,

- (1) when P is formed by non-hierarchical variables, iff P is true for o ;
- (2) when pr is a hierarchical variable and P is of the form $(pr = c)$, iff for value v of property pr in object o , $v \stackrel{\varepsilon}{=} c$ (if the value v of the object can be used instead of c with confusion ε);
- (3) when P is of the form $P_1 \vee P_2$, iff P_1 holds for o within ε or P_2 holds for o within ε ;
- (4) when P is of the form $P_1 \wedge P_2$, iff P_1 holds for o within ε and P_2 holds for o within ε ;
- (5) when P is of the form $\neg P_1$, iff P_1 does not hold for o within ε . ♦

Example 4.5 (refer to hierarchies H_1 and H_2 above): Let us define predicates $P = (\text{lives_in} = \text{USA}) \vee (\text{pet} = \text{cat})$, $Q = (\text{lives_in} = \text{USA}) \wedge (\text{pet} = \text{cat})$, $R = \neg (\text{lives_in} = \text{Spanish_Speaking_Island})$; and objects (Ann (lives_in USA) (pet snake)), (Bill (lives_in English_Speaking_Island) (pet citric)), (Fred (lives_in USA) (pet cat)), (Tom (lives_in Mexico) (pet cat)), (Sam (lives_in Cuba) (pet pine)). Then we have the following results (Table 2).

¹¹ For non-hierarchical variables, a match in value means $\text{conf} = 0$; a mismatch means $\text{conf} = \infty$.

4.2.2 The smallest ε for which $P(o)$ is true

How close is Tom to be like Ann in Example 4.5 of §4.2.1? Ann lives in the USA and her pet is a snake, while Tom lives in Mexico and his pet is a cat. When we apply $S = (\text{lives_in} = \text{USA}) \wedge (\text{pet} = \text{snake})$ to Tom¹², we see that S starts holding for $\varepsilon=1$. The answer to “How close is Tom to Ann?” is 1. Now we ask: How close is Ann to Tom? Ann is close to Tom starting from $\varepsilon=2$; that is, $(\text{lives_in} = \text{Mexico}) \wedge (\text{pet} = \text{cat})$ does not hold for Ann at $\varepsilon=1$, but it starts holding for her at $\varepsilon=2$. This hints how to define the “closeness to,” a number (a confusion) between an object o and a predicate P .

Definition 4.7. Object o ε -fulfills predicate P at threshold ε , if ε is the smallest number for which P holds for o within ε . ♦

Definition 4.8. Such smallest ε from Definition 4.7 is the closeness of o to P ¹³. ♦

Closeness is an integer number defined between an object and a predicate. The closer is ε to 0, the “tighter” P holds. Compare with the *membership function* for fuzzy sets.

4.3 Confusion between variables (not values) that form a hierarchy

What could be the error in “Sue directed the thesis of Fred”, if all we know is “Sue was in the examination committee of Fred”? Up to now, the *values* of a hierarchical variable form a hierarchy. Now, consider the case where the *variables* (or relations) form a hierarchy.

For instance, relative and brother, in a universe of kinship relations $E = \{\text{sister, aunt...}\}$. Consider hierarchies H_3 and H_4 :

¹² Here predicate $S = (\text{lives_in} = \text{USA}) \wedge (\text{pet} = \text{snake})$ represents “to be like Ann.” That is, from the definition of Ann (in example 4.5), we construct predicate S . Similarly, predicate $(\text{lives_in} = \text{Mexico}) \wedge (\text{pet} = \text{cat})$ embodies “to be like Tom” (see Tom in example 4.5).

¹³ The relation ‘closeness’ is not symmetric.

(Hierarchy H₃) relative \propto {close_relative \propto {father mother son daughter brother sister}
mid_relative \propto {aunt uncle niece cousin} far_relative \propto {grandfather grandmother grandson grand-
daughter grandaunt granduncle grandcousin grandniece} };

(Hierarchy H₄) player \propto {soccer_player \propto {John Ed} basketball_player \propto {Susan Fred} }.

In hierarchy H₃, $\text{conf}(son, relative) = 0$; $\text{conf}(relative, son) = 2$. We know that, for object (Kim (close_relative Ed) (pet cat)), the predicate $V = (\text{close_relative Ed})$ holds with confusion 0. It is reasonable to assume that $W = (\text{son Ed})$ holds for Kim with confusion 1¹⁴; that $X = (\text{relative Ed})$ holds for Kim with confusion 0. Moreover, since Ed is a member of hierarchy H₄, it is reasonable to assume that for object (Carl (close_relative soccer_player) (pet pine)) the predicate V holds with confusion 1, X holds with confusion $0+1 = 1$ and W holds with confusion $1+1 = 2$.

Thus, we can extend the definition of queries to variables that are members of a hierarchy, by adding another line to Definition 4.6 of §4.2.1, thus:

- (6) when P is of the form $(var = c)$, for var a variable member of a hierarchy, iff \exists variable var_2 for which $(var_2 = c)$ holds for o within $\varepsilon - \text{conf}(var_2, var)$, where var_2 also belongs to the hierarchy of var . ♦

The confusion of the variables *adds* to the confusion of the values.

Example 4.6: For (Burt (relative basketball_player) (pet cat)), V holds with confusion $1+2=3$, W with confusion $2+2=4$, and X with confusion $2+0=2$.

¹⁴ We are looking for a person that is a son of Ed, and we find Kim, a close relative of Ed.

4.4 Objects' similarity and accumulated confusion.

The three hierarchies of Figures 2-4 will allow us to introduce other new concepts such as identical, substitute, similar, etc. and accumulated confusion.

4.4.1 Identical, very similar, somewhat similar objects.

Objects are entities described by k (property, value) pairs, which in our notation we refer to as (variable, value) pairs. They are also called (relationship, attribute) pairs in databases. An object o with k (variable, value) pairs is written as $(o (v_1 a_1) (v_2 a_2) \dots (v_k a_k))$.

Example 4.7: (Bob (travels-by boat) (owns bird) (weighs heavy)).

We want to estimate the error in using object o' instead of object o . For an object o with k (perhaps hierarchical) variables v_1, v_2, \dots, v_k and values a_1, a_2, \dots, a_k , we say about another object o' with same variables v_1, v_2, \dots, v_k but with values a'_1, a'_2, \dots, a'_k we have the following definitions:

Definition 4.9. o' is *identical* to o , if $a'_i = a_i$ for all $1 \leq i \leq k$. All corresponding values are identical. ♦

If all we know about o and o' are their values on variables v_1, v_2, \dots, v_k and both objects have these values pair wise identical, then we can say that “for all we know,” o and o' are the same.

Definition 4.10. o' is a *substitute* for o , if $\text{conf}(a'_i, a_i) = 0$ for all $1 \leq i \leq k$. ♦

There is no confusion between a value of an attribute of o' and the corresponding value for o . We can use o' instead of the intended o with confusion 0.

Definition 4.11. o' is *very similar* to o , if $\sum_i \text{conf}(a'_i, a_i) = 1$. ♦

The confusions add to 1.

Definition 4.12. o' is *similar* to o , if $\sum_i \text{conf}(a'_i, a_i) = 2$. ♦

Definition 4.13. o' is *somewhat similar* to o , if $\sum_i \text{conf}(a'_i, a_i) = 3$. ♦

Definition 4.14. In general, o' is *similar_n* to o , if $\sum_i \text{conf}(a'_i, a_i) = n^{15}$. ♦

4.4.2 Accumulated confusion

For compound predicates, a tighter control of the inaccuracy or confusion is possible if we require that the accumulated mismatch does not exceed a threshold ε . This is accomplished by the following definition.

Definition 4.15. P holds for object o with accumulated confusion ε , written P^ε holds for o ,

- when P^ε is formed by non-hierarchical variables, iff P is true for o ,
- when pr is a hierarchical variable and P^ε is of the form $(pr=c)$, iff for value v of property pr in object o , $v =_{\varepsilon} c$. [That is, if the value v can be used instead of c with confusion ε],
- when P^ε is of the form $P_1 \vee P_2$, iff P_1^ε holds for o or P_2^ε holds for o ,
- when P^ε is of the form $P_1 \wedge P_2$, iff there exist confusions a and b such that $a+b \leq \varepsilon$ and P_1^a holds for o and P_2^b holds for o ,
- when P^ε is of the form $\neg P_1$, iff P_1^ε does not hold for o . ♦

4.4.3 Confusion between sets of qualitative values

An object (§4.2.1) may have a property (such as *owns*) that appears several times. Thus, (John *owns* citric) (John *owns* snake). To compare them, we need the following:

Definition 4.16. The confusion between two sets of qualitative values belonging to the same hierarchy is defined as $\text{conf}(\{a_1, a_2, \dots, a_m\}, \{b_1, b_2, \dots, b_n\}) = \text{MIN}_{1 \leq i \leq m, 1 \leq j \leq n} \text{conf}(a_i, b_j)$. ♦

Example 4.8: Let $A = \{\text{freedom, justice, social group}\}$, $B = \{\text{development, social group}\}$, $\text{conf}(A, B)$ is $\text{MIN}(\{\text{conf}(\text{freedom, development}), \text{conf}(\text{freedom, social group}), \text{conf}(\text{justice, development}), \text{conf}(\text{justice, social group}), \text{conf}(\text{social group, development}), \text{conf}(\text{social group, social group})\})$. Because $\text{conf}(\text{social group, social group})=0$ and 0 is the lowest confusion value, $\text{conf}(A, B)=0$.

¹⁵ Relations introduced in this section are not symmetric.

5 Querying a Database with Predicates that Are Imperfectly Fulfilled

The constructs of Section 4 are used in an extended SQL notation to retrieve from a database answers imperfectly fulfilling a predicate up to a desired confusion (error). The extended SQL expression (a query) is automatically transformed to normal SQL, which then retrieves. It is shown how to extend *any* relational database for precision-controlled retrieval and updates. This section abridges the work [46].

5.1 Extended SQL

Definition 5.1 (Extended SQL). To query with controlled precision a table T of a database, SQL is extended by the following constructs:

- $\text{conf}(R, s) \leq \varepsilon$, a SQL representation for $(R=s)_\varepsilon$, is a condition procedure¹⁶ used in a WHERE or HAVING clause, which is true iff $\text{conf}(r, s) \leq \varepsilon$. R is the name of a column of T that is a hierarchical variable (a variable or column having hierarchical values), r is each of these values, and s is the intended or expected qualitative value. ♦

Example 5.1: An example of retrieval using confusion appears in Figure 5.

Example 5.2: $\text{conf}(\text{cars.type}, \text{suv}) \leq 0$ represents in extended SQL the predicate $(\text{type} = \text{suv})_0$ and will select rows from Figure 6 whose type is SUV (Sport Utility Vehicle) with confusion 0; that is, all rows where $(\text{type} = r)$ and $\text{conf}(r, \text{suv}) \leq 0$. It returns rows 2 and 7.

- $\text{conf}(R)$ is a SQL expression [a shorthand for $\text{conf}(R, s)$], used in ‘SELECT $\text{conf}(R)$ ’, or ‘GROUP BY $\text{conf}(R)$ ’ or ‘ORDER BY $\text{conf}(R)$ ’, which returns for each row of table T, $\text{conf}(R, s)$. ♦

¹⁶ $(R=s)$ is a predicate, for instance $(\text{address}=\text{USA})$; adding confusion we have $(R=s)_\varepsilon$.

That is, $\text{conf}(R)$ returns for table T a list of numbers corresponding to the confusion of the value of property R for each row of T .

Example 5.3: see Figure 7.

5.2 Querying with extended SQL

Definition 5.2 (Writing queries in extended SQL). The algorithm $\text{EXPR} = \text{replace}(P)$ to replace a precision-controlled predicate P by its equivalent extended SQL expression EXPR is:

- $(R = s)_\varepsilon$ should be replaced by `'conf(' R ', ' s ') ≤' ε`, when R is the name of a column of a table, and s a symbolic value.
- $(P_1 \vee P_2)_\varepsilon$ should be replaced by `(' replace($P_{1\varepsilon}$) 'OR' replace ($P_{2\varepsilon}$) ')'`.
- $(P_1 \wedge P_2)_\varepsilon$ should be replaced by `(' replace($P_{1\varepsilon}$) 'AND' replace ($P_{2\varepsilon}$) ')'`.
- $\neg P$ should be replaced by `'NOT (' replace (P) ')'`.
- $(P_1 \wedge P_2)^\varepsilon$ should be replaced by `('replace(P_1) ' AND ' replace(P_2) ' AND (conf(' P_1 ') +conf(' P_2 ')) ≤' ε ')'`. ♦

Example 5.4: $(\text{type} = \text{suv})_3 \wedge [(\text{engine} = \text{4-cylinder gas})_2 \vee (\text{interior} = \text{fabric})_1]_1$ is replaced by `conf(type, suv) ≤ 3 AND (conf(engine, 4-cylinder gas) ≤ 2 OR conf(interior, fabric) ≤ 1)`.

Example 5.5: Henry wishes to buy a car with the following features: a SUV, having a 4-cylinder diesel engine with leather car's interiors. This means: $(\text{cars.type} = \text{suv})_2 \wedge (\text{cars.engine} = \text{4-cylinder gas})_2 \wedge (\text{cars.interior} = \text{leather})_2$. This is replaced by `conf(type, SUV) ≤ 2 AND conf(engine, '4-cylinder diesel') ≤ 2 AND conf(interior, leather) ≤ 2)`. Henry calls to the local car's store, giving the requested features to Vilma, the seller. She has the cars listed in Figure 6. The

result is given in Figure 7.

Given the nearest options of a car for Henry, he is finally interested in the Volkswagen Eurovan because it is the most similar to the desired car but also to the Ford Ecosport, not too similar but cheaper than the Eco-sport. Henry makes an appointment with Vilma to physically check the selected cars.

5.3 Queries: retrieving objects that match P_{ϵ}

Example 5.6: The Wall-Man supermarket will open a new branch in Los Angeles. A manager or an experienced worker to be promoted has to be assigned. The desired worker is described as a worker with a degree in business administration, living close to Los Angeles, with a position as manager and excellent qualifications. Wall-Man's workers are shown in Figure 8. Figure 9 shows the hierarchies used for each column:

Sorting the workers to choose the best one to be assigned on the new branch is made with only one query with confusion. The query is made writing the constraints in terms of confusion as $(worker.degree, business\ administration)_3 \wedge (worker.location, los\ angeles)_3 \wedge (worker.position, manager)_3 \wedge (worker.qualification, excellent)_3$. This is replaced by `conf(worker.degree, business administration) ≤ 3 AND conf(worker.location, los angeles) ≤ 3 AND conf(worker.position, manager) ≤ 3 AND conf(worker.qualification, excellent) ≤ 3`). The enriched SQL and result is shown in Figure 10.

Rosa Lopez is the best candidate to occupy the manager position on the new branch in Los Angeles.

Example 5.7: Confusion can be used to update rows. Because of a new Californian law, Wall-Man has to apply a tax of 0.1% to the salary of workers in California. Updating with standard SQL is a tedious process (every item has to be updated by hand). Using extended SQL with one query is shown in Figure 11.

Example 5.8: The Amazonic enterprise sells books on the web. An economist (Amartius) wishes to sell his brand new book focused on clients that would be interested in the contents. Amartius describes his book in

terms of qualitative values. Using a database of Amazonic's clients he wants to find clients having preferences close to the themes of the book, which are development, market, poverty, freedom, wto, world bank, government and social groups. If a client has interests in at least one of the themes of interest of the book, he is a potential buyer. The description of Amartiu's book in terms of confusion is: $\text{conf}(\text{interest}, \text{development}) \leq 2$ OR $\text{conf}(\text{interest}, \text{market}) \leq 2$ OR $\text{conf}(\text{interest}, \text{poverty}) \leq 2$ OR $\text{conf}(\text{interest}, \text{freedom}) \leq 2$ OR $\text{conf}(\text{interest}, \text{wto}) \leq 2$ OR $\text{conf}(\text{interest}, \text{world bank}) \leq 2$ OR $\text{conf}(\text{interest}, \text{government}) \leq 2$ OR $\text{conf}(\text{interest}, \text{social groups}) \leq 2$).

Figure 12 lists clients; Figure 13 shows the hierarchy for interests, and Figure 14 shows the result of the query.

6 Conclusion

The notions of hierarchy and hierarchical variable are used to measure the *confusion* when a symbolic value is used instead of another (the intended, or correct, value). This creates a natural generalization for predicates and queries. *Confusions* were introduced and developed for hierarchies formed by sets, bags, and lists, and they were extended (§4.1.3) to mixed hierarchies too.

The concepts herein developed have practical applications, since they mimic the manner in which people process qualitative values. Predicates with controlled precision¹⁷ $P_{\epsilon}(o)$ (called “ P holds for o with confusion ϵ ”) and $P^{\epsilon}(o)$ (called “ P holds for o with accumulated confusion ϵ ”) allow us to perform precision-controlled retrieval of hierarchical values. They permit “loose retrieval” (retrieval with defined confusion bounds) of objects that sit in a database. By §4.3, *relations* (columns in a database¹⁸) could also be “sloppily” specified. Moreover, such database could be an existing “normal” database (where no precision-controlled retrieval was

¹⁷ The precision is controlled through the confusion ϵ .

¹⁸ Any database is a relational one, because a datum makes sense only within a context.

defined), to which one or more definitions of hierarchies are attached. This in fact provides a procedure (a “kit” [46]) to extend *any* (existing) database to another in which imprecise retrievals are possible. Furthermore, this extension can be done without recompiling application programs. Old programs (with no precision retrieval) still work as before, whereas new application programs can exploit the “normal” database as if it were precision-controlled. In fact, a “normal” database now becomes a “precision-controlled” database when the extension (the kit) is applied to it.

References

- [1] V. Alexandrov and S. Levachkine, “Cognitive promptings for semantic-mind analysis and object-oriented data integration of information flows”. In: Levachkine, S., Serra, J., and Egenhofer, M. (eds.), *Proc. Int. Workshop Semantic Processing of Spatial Data (GEOPRO 2003)*, Research on Computing Science, Vol. 4 (2003) 2-10
- [2] N. Guarino, “Formal ontology, conceptual analysis, and knowledge representation”, *Int. J. Human and Computer Studies*, Vol. 43 (1995) 625-640
- [3] A. Sheth, “Changing focus on interoperability in information systems: From system, syntax, structure to semantics”. In: Goodchild, M., Egenhofer, M., Fegeas, R., and Kottman, C. (eds.), *Interoperating Geographic Information Systems* (1999) 5-30
- [4] A. Sheth and V. Kashyap, “So far (schematically) yet so near (semantically)”. In: Hsiao, D., Neuhold, E., and Sacks-Davis, R. (eds.), *Proc. IFIP WG2.6 Database Semantics Conf. Interoperable Database Systems*, Vol. DS-5 (1992) 283-312
- [5] N. Guarino, C. Masolo, and G. Verete, “OntoSeek: Content-based access to the web”, *IEEE J. Intelligent Systems*, Vol. 14 (1999) 70-80
- [6] E. Voorhees, “Using WordNet for text retrieval”, In: Fellbaum C. (ed.), *WordNet: An Electronic Lexical Database*, Cambridge, Mass.: The MIT Press (1998) 285-303
- [7] J. Jiang and D. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy”, *Proc. Int. Conf. Computational Linguistics (ROCLING X)* (1997)
- [8] A. Smeaton and I. Quigley, “Experiment on using semantic distance between words in image caption retrieval”,

- Proc. 19th Int. Conf. Research and Development in Information Retrieval (SIGIR 1996)* (1996)
- [9] J. Lee, M. Kim, and Y. Lee, "Information retrieval based on conceptual distance in IS-A hierarchies", *J. Documentation*, Vol. 49 (1993) 188-207
- [10] A. Goni, E. Mena, and A. Illarramendi, "Querying heterogeneous and distributed data repositories using ontologies". In Charrel, P.-J. and Jaakkola, H. (eds.), *Information Modeling and Knowledge Base IX*, IOS Press (1998) 19-34
- [11] S. Levachkine and A. Guzman-Arenas, "Hierarchies measuring qualitative variables", *Lecture Notes in Computer Science*, Vol. 2945 (2004) 258-270
- [12] V.P. de Gyves and A. Guzman-Arenas, "A distributed digital text accessing and acquisition system: BiblioDigital", *Lecture Notes in Computer Science*, Vol. 3061 (2004) 274-283
- [13] N. Guarino, "Formal ontology in information systems". In: Guarino, N. (ed.), *Proc. 1st Int. Conf. Formal Ontology in Information Systems*, Springer-Verlag (1998) 3-15
- [14] Y. Bishr, "Semantic aspects of interoperable GIS", Wageningen Agricultural Univ. and ITC, The Netherlands (1997)
- [15] M. Bright, A. Hurson, and S. Pakzad, "Automated resolution of semantic heterogeneity in multidatabases", *ACM Trans. Database Systems*, Vol. 19 (1994) 212-253
- [16] P. Fankhauser and E. Neuhold, "Knowledge based integration of heterogeneous databases", In: Hsiao, D., Neuhold, E., and Sacks-Davis, R. (eds.), *Proc. IFIP WG2.6 Database Semantics Conf. Interoperable Database Systems*, Vol. DS-5 (1992) 155-175
- [17] C. Collet, M. Huhns, and W. Shen, "Resource integration using a large knowledge base in Carnot", *Computer*, Vol. 24 (1991) 55-62
- [18] A. Tversky, "Features of similarity", *Psychological Rev.*, Vol. 84 (1977) 327-352
- [19] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity and natural language", *J. Artificial Intelligence Research*, Vol. 11 (1999) 95-130
- [20] S. Ross, *A first course in probability*. New York: Macmillan, 1976
- [21] E. Mena, V. Kashyap, and A. Sheth, "OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies", *Proc. Int. Conf. Cooperative Information Systems (CoopIS 1996)* (1996)
- [22] V. Kashyap and A. Sheth, "Semantic heterogeneity in global information systems: The role of metadata, context, and

- ontologies". In: Papazoglou, M. and Schlageter, G. (eds.), *Cooperative Information Systems: Tends and Directions*, (1998) 139-178
- [23] B. Bergamaschi, S. Castano, S. De Capitani di Vermercati, S. Montanari, and M. Vicini, "An intelligent approach to information integration". In: Guarino, N. (ed.), *Proc. 1st Int. Conf. Formal Ontology in Information Systems*, Springer-Verlag (1998) 253-268
- [24] A. Gangemi, D. Pisanelli, and G. Steve, "Ontology integration: Experiences with medical terminologies". In: Guarino, N. (ed.), *Proc. 1st Int. Conf. Formal Ontology in Information Systems*, Springer-Verlag (1998) 163-178
- [25] W. Kim and J. Seo, "Classifying schematic and data heterogeneity in multidatabase systems", *Computer*, Vol. 24 (1991) 12-18
- [26] P. Visser, D. Jones, T. Bench-Capon, and M. Shave, "Assessing heterogeneity by classifying ontology mismatches". In: Guarino, N. (ed.), *Proc. 1st Int. Conf. Formal Ontology in Information Systems*, Springer-Verlag (1998) 148-162
- [27] P. Weinstein and P. Birmingham, "Comparing concepts in differentiated ontologies", *Proc. 12th Int. Workshop Knowledge Acquisition, Modeling, and Management*, (1999)
- [28] E. Mena, V. Kashyap, and A. Sheth, "OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies", *Distributed and Parallel Databases*, Vol. 8 (2000) 223-271
- [29] A. Rodriguez and M. Egenhofer, "Determining semantic similarity among entity classes from different ontologies", *IEEE Trans. Knowledge and Data Engineering*, Vol. 15, No. 2 (2003) 442-456
- [30] N.T. Bhin, A.M. Tjoa, and R. Wagner, "Conceptual multidimensional data model based on meta-cube", *Lecture Notes in Computer Science*, Vol.1909 (2000) 24-31
- [31] A. Guzman-Arenas, "Finding the main themes in a Spanish document", *J. Expert Systems with Applications*, Vol. 14, No.1/2 (1998) 139-148
- [32] A. Guzman-Arenas and J. Olivares-Ceja, "Finding the most similar concepts in two different ontologies", *Lecture Notes in Artificial Intelligence*, Vol. 2972 (2004) 129-138
- [33] J. Everett and D. Bobrow, "Making ontologies work for resolving redundancies across documents", *Comm. ACM*, Vol. 45, No. 2 (2002) 55-60
- [34] A. Gelbukh, G. Sidorov, and A. Guzman-Arenas, "Document comparison with a weighted topic hierarchy", *Proc. 10th Int. Conf. Database and Expert System applications: Workshop Document Analysis and Understanding for*

Document Databases (DEXA 1999) (1999) 566-570

- [35] M. Montes-y-Gómez, A. Lopez-Lopez, and A. Gelbukh, “Information retrieval with conceptual graph matching”, *Lecture Notes in Computer Science*, Vol. 1873 (2000) 312-321
- [36] D. Lin, “An information-theoretic definition of similarity”. *Proc. 15th Int. Conf. on Machine Learning* (ICML 1998)
- [37] P. Resnik, “Disambiguating noun groupings with respect to WordNet senses”, In: Armstrong, S. *et al.* (eds.), *Natural Language Processing Using Very Large Corpora*, Kluwer Academic Publishing: Dordrecht (1995) 77-98
- [38] A. Budanitsky and G. Hirst, “Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures”. *Proc. North American Chapter of the Association for Computational Linguistics* (NAACL-2000), Pittsburgh, PA.
- [39] N. Noy, R.W. Ferguson, and M.A. Musen, “The knowledge model of Protégè-2000: Combining interoperability and flexibility”, *Stanford Medical Informatics Technical Report*, Stanford Univ. (2000)
- [40] D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems*. Addison-Wesley, 1989
- [41] A. Guzman-Arenas and S. Levachkine, “Graduated errors in approximate queries using hierarchies and ordered sets”, *Lecture Notes in Artificial Intelligence*, Vol. 2972 (2004) 139-148
- [42] A. Guzman, J. Olivares, A. Demetrio, and C. Dominguez, “Interaction of purposeful agents that use different ontologies”, *Lecture Notes in Artificial Intelligence*, Vol. 1793 (2000) 557-573
- [43] J. Olivares-Ceja and A. Guzman-Arenas, “Concept similarity measures the understanding between two agents”, *Lecture Notes in Computer Science*, Vol. 3136 (2004) 182-194
- [44] F. Martinez-Trinidad and A. Guzman-Arenas, “The logical combinatorial approach to Pattern Recognition, an overview through selected works”, *Pattern Recognition*, Vol. 34 (2001) 741-751
- [45] J.-C. Simon. *Patterns and operators: The foundations of data representation*. McGraw-Hill, 1984
- [46] V. P. de Gyves, A. Guzmán-Arenas, and S. Levachkine, “Extending Databases to Precision-Controlled Retrieval of Qualitative Information”, *ISSADS 2005*, 21-32

LIST OF FIGURES

- Figure 1: A **hierarchy H_2** of live beings. $\text{conf}(\text{cat}, \text{mammal})= 0$; $\text{conf}(\text{mammal}, \text{cat})= 1$.
- Figure 2: A **hierarchy H_5** of vehicles. Some qualitative values, like compact, represent sets: {sport compact, standard compact, family compact} in our example.
- Figure 3: A **hierarchy H_6** having some ordered sets: (4-cylinder gas < 6-cylinder gas < 8-cylinder gas), (4-cylinder diesel < 6-cylinder diesel < 8-cylinder diesel). Note that gas and diesel are not members of an ordered set.
- Figure 4: A **hierarchy H_7** of automobile's interiors.
- Figure 5: Query $(\text{car.type} = \text{pick-up})_2$ returns cars of type pick-up with confusion 2. Hierarchy H_5 of Figure 2 is used.
- Figure 6: Table of cars.
- Figure 7: Querying, sorting, and showing values for $(\text{type} = \text{suv})_1 \wedge (\text{cars.engine} = \text{4-cylinder gas})_2 \wedge (\text{cars.interior} = \text{leather})_2$.
- Figure 8: Wall-Man's workers on a database table.
- Figure 9: Hierarchies for the qualitative values of the table of workers in Wall-Man.
- Figure 10: Sorting workers to choose the best one to manage a new branch.
- Figure 11: Using extended SQL to update a database. This is not just made by one query but faster to the database server to process it.
- Figure 12: Clients of Amazonic and their interests. Notice that John Doe has many interests.
- Figure 13: Hierarchy of readers' interests for Amazonic.
- Figure 14: Sorting Wall-Man's bests workers to occupy a new manager position. This show confusion between objects with many qualitative values for a property, as seen in §4.4.3.

LIST OF TABLES

Table 1: $\text{conf}(r, s)$: confusion in using r (a certain row) instead of s (a certain column) for the live beings of H_2 .

Table 2: How the predicates P , Q and R of example 4.5 hold for several objects.



Figure 1: A hierarchy H_2 of live beings. $\text{conf}(\text{cat}, \text{mammal})=0$; $\text{conf}(\text{mammal}, \text{cat})=1$.

```
types{
  suv{
    truck
    cargo vehicle{
      pick-up
      compact pick-up
      minivan }
    sport suv }
  compact{
    sport compact
    standard compact
    family compact }
  luxury{
    luxury sport
    large range
    mid range
  }
  economic }
```

Figure 2: A hierarchy H_5 of vehicles. Some qualitative values, like compact, represent sets: {sport compact, standard compact, family compact} in our example.

```
engine{
  gas*{
    4-cylinder gas
    6-cylinder gas
    8-cylinder gas
  }
  diesel*{
    4-cylinder diesel
    6-cylinder diesel
    8-cylinder diesel
  }
}
```

Figure 3: A hierarchy H_6 having some ordered sets: (4-cylinder gas < 6-cylinder gas < 8-cylinder gas), (4-cylinder diesel < 6-cylinder diesel < 8-cylinder diesel). Note that gas and diesel are not members of an ordered set.


```
interior{
  leather{
    traditional leather
    vinyl coated leather
  }
  standard{
    polyurethane
    fabric
  }
}
```

Figure 4: A hierarchy H_7 of automobile's interiors.

```
Select
cars.manufacturer,cars.model,cars.type,  conf(cars.type)  from
cars  where  conf(cars.type,  pick-up)  ≤  2  order  by
conf(cars.type)
```

MANUFACTURER	MODEL	TYPE	conf (type,pick-up)
Nissan	PU 2.4	compact pick-up	0
Volkswagen	Eurovan	minivan	0
Ford	F-150	pick-up	0
Jeep	Wrangler	suv	2
Jeep	Grand Cherokee	truck	2
Hummer	H3	truck	2
Nissan	X-Terra	truck	2
Volkswagen	Touareg	truck	2
Ford	Ecosport	truck	2

Figure 5: Query $(car.type = pick-up)_2$ returns cars of type pick-up with confusion 2. Hierarchy H_5 of Figure 2 is used.

manufacturer	model	type	engine	interior
Ford	Focus	standard compact	4-cylinder gas	traditional leather
Jeep	Wrangler	suv	4-cylinder gas	fabric
Jeep	Grand Cherokee	truck	8-cylinder gas	polyurethane
BMW	320s	mid range	8-cylinder gas	vinyl coated leather
Ford	F-150	pick-up	6-cylinder gas	traditional leather
Nissan	PU 2.4	compact pick-up	4-cylinder gas	fabric
Volkswagen	Eurovan	minivan	4-cylinder diesel	polyurethane
Hummer	H3	truck	8-cylinder diesel	vinyl coated leather
Chevrolet	Cavalier	family compact	4-cylinder gas	traditional leather
Nissan	Tsuru	economic	4-cylinder gas	fabric
Nissan	X-Terra	truck	4-cylinder gas	polyurethane
Volkswagen	Touareg	truck	6-cylinder gas	traditional leather
Ford	Ecosport	truck	4-cylinder gas	traditional leather
Chrysler	300	large range	8-cylinder gas	vinyl coated leather

Figure 6: Table of cars.

```

Select
cars.manufacturer,cars.model,cars.price,conf(cars.
type)+conf(cars.engine)+conf(cars.interior) as
SUM_CONFUSION from cars where ( conf(cars.type,
suv ) ≤ 3 and conf(cars.engine, 4-cylinder diesel )
≤ 3 and conf(cars.interior, leather ) ≤ 3 ) order by
SUM_CONFUSION

```

MANUFACTURER	MODEL	PRICE	SUM_CONFUSION
Volkswagen	Eurovan	26000	1
Ford	F-150	23000	2
Ford	Ecosport	17000	2
Volkswagen	Touareg	54000	2
Nissan	PU 2.4	12000	3
Nissan	Tsuru	10000	3
Chevrolet	Cavalier	14000	3
Jeep	Wrangler	18000	3
Ford	Focus	15000	3
Nissan	X-Terra	22000	3
Jeep	Grand Cherokee	35000	3

Figure 7: Querying, sorting, and showing values for $(type = suv)_1 \wedge (cars.engine = 4-cylinder gas)_2 \wedge (cars.interior = leather)_2$.

NAME	POSITION	QUAL.	LOCATION	DEGREE
Martin Diaz	inventories	G	austin	[accounting high School]
Bruce Morris	supervisor	G	new york city	[industrial administration high school]
Joana Simpson	manager	N	san antonio	accounting college
Joseph Montoya	cash registerer	VP	anchorage	informatics college
Angel Diaz	cash registerer	E	new york city	[business administration college]
Joel Martinez	supervisor	E	san antonio	accounting college
Lee Huan	cash registerer	N	east la	architect
Rosa Lopez	supervisor	VG	east la	[accounting high School]
Emile Chandler	supervisor	G	sacramento	architect
Sonya Norman	pharmacy	B	downtown	doctor
Ramiro Carranza	manager	G	ontario	[industrial administration high school]
Armando Huerta	supervisor	N	los angeles	informatics college

Note: in QUAL. column, G = good, N = normal, B = bad, VP = very poor, VG = very good, and E = excellent. The TAX column is hidden in this list to save space.

Figure 8: Wall-Man's workers on a database table.

<p>Hierarchy for location</p> <pre> usa{ continental{ alaska{ anchorage } } non continental{ california{ los angeles{ downtown, east la } sacramento, on- tario } texas{ austin, san an- tonio } new york{ new york city, buffalo } } } </pre>	<p>Hierarchy for degree</p> <pre> degree{ enterprise{ informatics{ informatics college } accounting{ accounting high school, accounting college } administration{ business{ administration high school, business administration college } industrial{ industrial administration high school } } } medicine{ doctor, surgeon } engineery{ metalmechanics college, industrial co- llege, architect } } </pre>
<p>Hierarchy for quali- fication</p> <pre> qualifications*{ bad very poor poor normal good very good excellent } </pre>	<p>Hierarchy for position</p> <pre> position{ basic{ intence, food, tools, pharmacy } administration{ supervisor, manager } other{ cash registerer, attention to clients } } </pre>

Figure 9: Hierarchies for the qualitative values of the table of workers in Wall-Man.

```

select worker.*, ( conf(worker.position)+ conf(worker.qualification)
+conf(worker.location) +conf(worker.degree) as sum from worker where
conf(worker.position, 'manager' ) ≤ 3 and conf(worker.qualification,
'excellent' ) ≤ 3 and conf(worker.location, 'los angeles' ) ≤ 3 and
conf(worker.degree, 'business administration' ) ≤ 3 ) order by sum;

```

NAME	P	Q	LOCATION	DEGREE	SUM
Rosa Lopez	S	VG	east la	accounting high school	5
Ramiro Carranza	M	G	ontario	[industrial administration high School]	5
Joel Martinez	S	E	san antonio	accounting college	6
Armando Huerta	S	N	los angeles	informatics college	7
Bruce Morris	S	G	new york city	[industrial administration high School]	7
Joana Simpson	M	N	san antonio	accounting college	8

Note: In P (POSITION) column, S = supervisor, M = manager. In Q (QUALIFICATION) column, G = good, N = normal, B = bad, VG = very good, and E = excellent. TAX column is hidden for reasons of space.

Figure 10: Sorting workers to choose the best one to manage a new branch.

```

update worker set tax=0.001 where worker.name in (se-
lect      worker.name      from      worker      where
conf(worker.location, california ) ≤ 0 ) ;

select name, location, tax from worker;

```

NAME	LOCATION	TAX
Martin Diaz	austin	0
Bruce Morris	new york city	0
Joana Simpson	san antonio	0
Joseph Montoya	anchorage	0
Angel Diaz	new york city	0
Joel Martinez	san antonio	0
Lee Huan	east la	0.001
Rosa Lopez	east la	0.001
Emile Chandler	sacramento	0.001
Sonya Norman	downtown	0.001
Ramiro Carranza	ontario	0.001
Armando Huerta	los angeles	0.001

Figure 11: Using extended SQL to update a database. This is not just made by one query but faster to the database server to process it.

(1, John E. Doe, algebra) (1, John E. Doe, calculus) (1, John E. Doe, geometry) (2, Mary Lopez, languages) (2, Mary Lopez, spanish), (3, Ernest Smith, justice) (3, Ernest Smith, faith) (3, Ernest Smith, traditional) (4, Fredrich Miranda, moral) (4, Fredrich Miranda, ethics) (4, Fredrich Miranda, international organizations) (5, Josefina Wayne, money) (5, Josefina Wayne, elasticity), (5, Josefina Wayne, enterprise) (6, Luigi Fermi, traditional) (6, Luigi Fermi, contemporary) (7, Rosalia Martinez, freedom) (7, Rosalia Martinez, organizations) (8, Jose Gomez, economy) (8, Jose Gomez, traditional) (9, Ann Jackson, macroeconomics) (9, Ann Jackson, wto) (10, Masrtin Keyne, freedom) (10, Martin Keyne, justice) (10, Martin Keyne, social group)

Figure 12: Clients of Amazonic and their interests. Notice that John Doe has many interests.

```

interests{
  science{
    economy{
      macroeconomics{money, recession, unemployment, inflation}
      microeconomics{elasticity, scarcity, market, demand}
      development,
      poverty
    }
    mathematics{algebra, calculus, geometry}
    natural sciences{biology, ecology, chemical}
  }
  people{
    social values{moral, ethics, freedom, tradition, justice,
      faith}
    languages{english, spanish, french, chinese, portuguese}
    poetry{traditional, contemporary}
  }
  organizations{
    international organizations{wto, world bank, un, red cross,
      nato}
    general organizations{government, social group, enterprise}
  }
}

```

Figure 13: Hierarchy of readers' interests for Amazonic.

```

select ACIV.name, float4smaller( conf( ACIV.interest:development),
float4smaller(conf(ACIV.interest:market),float4smaller(conf(ACIV.inter
est:poverty),float4smaller(conf(ACIV.interest:freedom),float4smaller(c
onf(ACIV.interest:wto),float4smaller(conf(ACIV.interest:\"world
bank\"),float4smaller(conf(ACIV.interest:government), (conf(ACIV.intere
st:\"social_group\")))))))))) as conf_interest from ama-
zonic_clients_interests_view.interest as ACIV where
(conf(ACIV.interest, development ) ≤ 2 OR conf(ACIV.interest, market ) ≤
2 OR conf(ACIV.interest, poverty ) ≤ 2 OR conf(ACIV.interest, freedom )
≤ 2 OR conf(ACIV.interest, wto ) ≤ 2 OR conf(ACIV.interest, "'world
bank'" ) ≤ 2 OR conf( ACIV.interest, government ) ≤ 2 OR
conf(ACIV.interest, "'social_group'" ) ≤ 2 ) order by conf_interest

```

ID	CLIENT	INTEREST	CONF_INTEREST
7	Rosalia Martinez	freedom	0
10	Martin Keyne	freedom	0
9	Ann Jackson	wto	0
10	Martin Keyne	social_group	0
8	Jose Gomez	economy	1
9	Ann Jackson	macroeconomics	1
5	Josefine Wayne	money	1
5	Josefine Wayne	elasticity	1
4	Fredrich Miranda	moral	1
4	Fredrich Miranda	ethics	1
3	Ernest Smith	justice	1
10	Martin Keyne	justice	1
3	Ernest Smith	faith	1
4	Fredrich Miranda	international organizations	1
5	Josefine Wayne	enterprise	1
1	John E. Doe	algebra	2
1	John E. Doe	calculus	2
1	John E. Doe	geometry	2
2	Mary Lopez	languages	2
2	Mary Lopez	spanish	2
3	Ernest Smith	traditional	2
6	Luigi Fermi	traditional	2
8	Jose Gomez	traditional	2
6	Luigi Fermi	contemporary	2
7	Rosalia Martinez	organizations	2

Several clients appear repeated in the result with different interests and confusion values. Confusion between a set of client's interests and the set of themes of the book is the smallest of the confusion values for that client in the result. Conf(INTERESTS(Martin Keyne), INTERESTS(book)) = 0 according to 3.4.3, because the smallest confusion in the result for Martin Keyne is 0.

Figure 14: Sorting Wall-Man's best workers to occupy a new manager position. This shows confusion between objects with many qualitative values for a property, as seen in §4.4.3

	S								
	Live b.	Animal	Plant	Mam.	Snake	Citric	Pine	Cat	Lemon
Live b.	0	1	1	2	2	2	2	3	3
Animal	0	0	1	1	1	2	2	2	3
Plant	0	1	0	2	2	1	1	3	2
Mam.	0	0	1	0	1	2	2	1	3
Snake	0	0	1	1	0	2	2	2	3
Citric	0	1	0	2	2	0	1	3	1
Pine	0	1	0	2	2	1	0	3	2
Cat	0	0	1	0	1	2	2	0	3
Lemon	0	1	0	2	2	0	1	3	0

Table 1: $\text{conf}(r, s)$: confusion in using r (a certain row) instead of s (a certain column) for the live beings of H_2 .

	<i>P</i> holds within ϵ for:	<i>Q</i> holds within ϵ for:	<i>R</i> holds within ϵ for:
$\epsilon = 0$	Ann, Fred, Tom	Fred	Ann, Bill, Fred, Tom
$\epsilon = 1$	Ann, Fred, Tom	Fred, Tom	Ann, Fred, Tom
$\epsilon = 2$	Ann, Fred, Tom, Sam	Ann, Fred, Tom	Nobody

Table 2: How the predicates P , Q and R of example 4.5 hold for several objects.