Hierarchies Measuring Qualitative Variables

Serguei Levachkine and Adolfo Guzmán-Arenas

Centre for Computing Research (CIC) - National Polytechnic Institute (IPN)
UPALMZ, CIC Building, 07738, Mexico City, MEXICO
palych@cic.ipn.mx, a.guzman@acm.org

Abstract. Qualitative variables take symbolic values, such as *hot, shoe, Europe or France*. Sometimes, the values may be arranged in layers or levels of detail. For instance, the variable *place_of_origin* takes as level-1 values *European, African*... as level-2 values *French, German*... as level-3 values *Californian, Texan*... The paper describes a hierarchy, a mathematical construct among these variables. The confusion resulting when using a value instead of another is defined, as well as the closeness to which object *o* fulfills predicate *P*. Other operations among and properties of hierarchical values are derived. Hierarchies are compared with ontologies. Hierarchies find use in measuring linguistic relatedness or similarity. Hierarchical variables abound and are commonly used, often with suggestive string values, without fully realizing or exploiting its properties. We deal with arbitrary hierarchies. Examples are given.

1 Introduction

A datum is a relational entity. Nothing is a datum itself; i.e. a context[1] is required. This thesis is especially true for qualitative data. Notice that many works on qualitative data processing usually omit the problem under consideration context. In contrast, we use the hierarchies to measure similarity and dissimilarity between qualitative values, attempting to keep the context. To some extent, the notion of hierarchy provides an adequate tool for qualitative data analysis, processing and classification, because the hierarchies encapsulate the (sometimes ordered) relations between partitions of the dataset and therefore easily maintain the problem context.

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 and similarities? How related or close are these words? Some preliminary definitions follow.

[1] The notion of context depends on particular environment (subject domain, representation space...) into which the data are embedded. In turn the relatedness between data elements depends on the context. For example, the *pale* and *beige* could be much closed (to indistinguishable) in one context while in another they should be far distanced. Subsequently this paper concerns not only with the problem to appropriately define the closeness of data elements but also to take into consideration the properties of the representation space. This can be observed as a context-oriented approach to qualitative data processing (see also §1.3).

Element set. A set[2] E whose elements are explicitly defined. ↓[3] *Example*: {*red, blue, white, black, pale*}.
Ordered set. An element set whose values are ordered by a < ("less than") relation. ↓ *Example*: {*very_cold, cold, warm, hot, very_hot*}.
Covering. K is a covering for set E if K is a set of subsets $s_i \leq$ E, such that $\cong s_i$ = E. ↓ Every element of E is in some subset $s_i \subset$ K. If K is not a covering of E, we can make it so by adding a new s_j to it, named "others", that contains all other elements of E that do not belong to any of the previous s_i.
Exclusive set. K is an exclusive set if $s_i \sim s_j = \therefore$, for every $s_i, s_j \subset$ K. ↓ Its elements are mutually exclusive. If K is not an exclusive set, we can make it so by replacing every two overlapping $s_i, s_j \subset$ K with three: $s_i - s_j$, $s_j - s_i$, and $s_i \sim s_j$.
Partition. P is a partition of set E if it is both a covering for E and an exclusive set.
Qualitative variable. A single-valued variable that takes symbolic values. ↓ Its value cannot be a set.[4] By symbolic we mean qualitative, as opposed to numeric, vector or quantitative variables.

A symbolic value v **represents** a set E, written v ∇ E, if v can be considered a name or a depiction of E. ↓ *Example*: *Pale* ∇ {*white, yellow, orange, beige*}.

1.1 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 $\cong_i \{r_i \mid e_i \nabla r_i\}$ = E (The union of all sets represented by the e_i is E). ↓ *Example* (Hierarchy H_1): for E = {*Canada, USA, Mexico, Cuba, Puerto_Rico, Jamaica, Guatemala, Honduras, Costa_Rica*}={a, b, c, d, e, f, g, h, i}, a hierarchy H_1 is {*North_America, Caribbean_Island, Central_America*}={H_1^1, H_1^2, H_1^3}, where *North_America* ∇ {*Canada, USA, Mexico*}; *Caribbean_Island* ∇ {*English_Speaking_Island, Spanish_Speaking_Island*}={H_1^{21}, H_1^{22}}; *English_Speaking_Island* ∇ {*Jamaica*}; *Spanish_Speaking_Island* ∇ {*Cuba, Puerto_Rico*}; *Central_America* ∇ {*Guatemala, Honduras, Costa_Rica*}.

Hierarchies make it easier to compare qualitative values belonging to the same hierarchy (§3), and even to different hierarchies (procedure sim in [11]).

A **hierarchical variable** is a qualitative variable whose values belong to a hierarchy (The data type of a hierarchical variable is hierarchy). ↓ *Example*: *place_of_origin* that takes values from H_1. Note: hierarchical variables are single-valued.

---

[2] Perhaps infinite, perhaps empty.
[3] The symbol ↓ means: end of definition.
[4] Variable, attribute and property are used interchangeably. An object may have an attribute (Ex: 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.

Thus, a value for *place_of_origin* can be *North_America* or *Mexico,* but not {*Canada, USA, Mexico*}, although *North_America* ∇ {*Canada, USA, Mexico*}.

1.2 Notation

The sets represented by each element of a hierarchy form a tree under the relation subset. *Example*: for H_1, such tree is given in Figure 1.

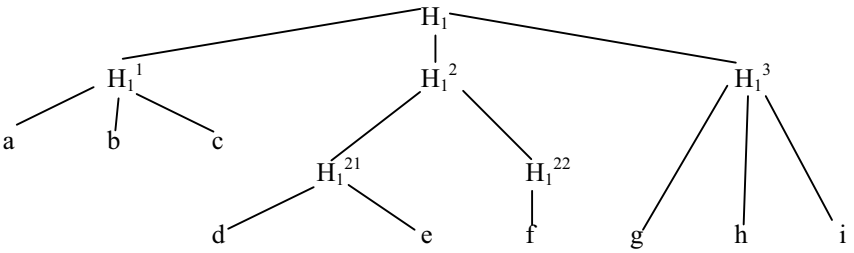

Fig. 1. The tree induced by hierarchy H_1.

We will also write a hierarchy such as H_1 thus: {*North_America* ∇ {*Canada USA Mexico*} *Caribbean Island* ∇ {*Spanish_Speaking_Island* ∇ {*Cuba Puerto_Rico*} *English_Speaking_Island* ∇{*Jamaica*} } *Central_America* ∇ {*Guatemala Honduras Costa_Rica*} }.

father_of (v). In a tree representing a hierarchy (such as H_1), the father_of a node is the node from which it hangs. ↓ Similarly, the **sons_of** (v) are the values hanging from v. The nodes with the same father are **siblings**. ↓ Similarly, **grand_father_of**, **brothers_of, aunt**, **ascendants**, **descendants**... are defined, when they exist. ↓ The **root** is the node that has no father.↓

1.3 Previous related work

CYC [6] was an early attempt to build the concept tree (an ontology) for common concepts. Clasitex [2] finds the themes of an article written in Spanish or English, performing a task equivalent to disambiguation of a word into its different senses. It uses the concept tree, and a word (words lie outside the context tree) *suggests the topic of* one or more concepts in the tree. A document that talks about Cervantes, horses and corruption will be classified (indexed) in these three nodes in the tree. In [3] [4], each agent possesses its own ontology of concepts, but must map these into natural language words for communication [11]. Thus LIA, a language for agent interaction [3], has an ontology comparator COM, that maps a concept from one ontology into the closest corresponding concept of another ontology. COM achieves communication without need of a common or *standard ontology;* it is used in sim of §3.4. Ontologies' relation to hierarchies will be further elaborated here.

The set of data items that we have to process is of course finite (Cf. footnote 1). First of all, we have to ask about the nature of the *representation space*, i.e., we need

to know whether the data can be regarded as "*values*" of certain "*variables*" (Cf. §1), and whether these variables have certain properties: are we at liberty to embed the data into some "*space*", and to perform certain *operations* on them?

Traditionally [12] [13], the representation space is regarded as a metric space with some "exotic" or *ad hoc* distance (e.g., ultrametric distance to measure the proximity among members of a hierarchy; see §2). However, this requires a proof that such a distance meets the needs of the classification problem under consideration. Since, in general, the data of a problem consist at best of distances in the ordinary sense, the requirement is to obtain the "exotic distance" from an "ordinary distance." The intermediate data conversion often makes it difficult for any algorithms to define and exploit errors in using one data element instead of another; this is crucial for many domains involving qualitative variables (§3). Another problem with this conversion is its significant computational cost. A solution for these problems herein developed is to avoid the requirement of the measure to be a "distance" (even an "exotic" distance), defining so-called similarity or dissimilarity (confusion) functions on data elements of arbitrary nature in a manner similar to the human handling of these qualitative variables (it is hard to expect that they first define a distance to distinguish the *low_cost* and *high_cost* of goods). This is the main goal of the present paper, its novelty and its unique contribution (§3).

2 Theoretical Background

In this section we put forward some formal definitions previously developed and extensively commented in [7] [9] [15]. We should underline that the notion of *ultrametric distance* introduced in the following (§2.3) is accepted as "natural" measure of the hierarchical elements [12] [13] but is useless as well as any other *distance* within our context-oriented approach. Thus, it should be revised and replaced in §3.

2.1 Partitions of a finite set

Two elements x and y of E are **equivalent** in a partition P if they belong to the same class s_i; this is denoted by xPy. ↓

Let **P**(E) be the set of all partitions of E; an **order relation** among the members of **P**(E), denoted by <, can be defined thus: for any two partitions P and P', P<P' iff xPy⇓ xP'y. Partition P is said to be **finer** than P'; it has more classes than P'.↓

A **lattice** structure for **P**(E) can be based on the order relation. For every pair of partitions P and P' there is a least upper bound (l.u.b.) P⟩ P', and greatest lower bound (g.l.b.) P⟨ P'. ↓

Let us call P_k a partition of k classes where k is the level of P_k. A partition P' is said to **cover** a partition P if and only if P' results from combining *two* classes of P. A **chain** in the lattice is a sequence of partitions in order, e.g. $(P_1,P_2,...,P_j)$ where $P_1< P_2 <...< P_j$; the term is understood in the sense of an elementary chain in graph theory.

2.2 Hierarchies

Let E be a set of n elements, ∩ (E) the set of all subsets of E and **P**(E) the **lattice** of the **partitions** defined by the **order relation** P < Q. Let CH be a complete **chain** in the lattice, i.e. a chain linking the **finest partition** P_n, of n elements, to the **coarsest partition** P_1=E. Now we can give two equivalent definitions of a **hierarchy**.

(1) A hierarchy is a set of partition classes constituting a complete chain, including in particular the set E itself and the n subsets formed by the elements of E. ↓

The passage from level k to level k-1 on CH corresponds to combining two classes. However, several levels can be passed over. Let P and Q be two non-consecutive partitions on CH, so that the classes of Q are either those of P or combinations of two or more classes of P. This leads to another direct definition.

(2) A hierarchy is a subset H of ∩ (E) such that (1) E⊂H, (2) if x and y are elements of E, then ″x∈″y∈H, (3) if h and h' are elements of H, then either h~ h'=∴ or h~ h'∏∴, in which case either h≤ h' or h'≤ h. ↓ *Example:* See Figure 1.

2.3 Ultrametrics

A partial ordering of the elements of a hierarchy can be based on the inclusion relation and can be made a total ordering by the process of ascending a complete chain CH. In general, the same hierarchy can be defined by several different chains; thus if E=″a,b,c,d,e,f∈ then for the hierarchy H formed by the subsets ″a∈ ″b∈ ″c∈ ″d∈ ″e∈ ″f∈with h_1=E, h_2=″a,b,c,d∈ h_3=″e,f∈and h_4=″a,b,c∈ we can use three chains CH1, CH2 and CH3, with their nodes numbered 0,1,2,3,4 as follows:

CH1	*a,b,c,d,e,f*	*abc,d,e,f*	*abc,d,ef*	*abcd,ef*	*abcdef*
CH2	*a,b,c,d,e,f*	*abc,d,e,f*	*abcd,e,f*	*abcd,ef*	*abcdef*
CH3	*a,b,c,d,e,f*	*a,b,c,d,ef*	*abc,d,ef*	*abcd,ef*	*abcdef*
	0	**1**	**2**	**3**	**4**

Two elements of E occur in the same subset at a given node of CH, this being a partition of E. Given the chain, the node numbers characterize each pair of elements of E. We can now show how they can be used to define a special kind of distance.

2.3.1 Ultrametric distance

If i, j and k are three elements of a set E, the **ultrametric distance** ι is d fined as a function of EΔE in R^+ as follows: ι (i,i)=0, ι (i,j)=ι (j,i), ι (i,j) Ωmax**Ψ**(i,k),ι (j,k)β ↓

So we might define a distance between elements of E by means of a chain of partitions, and it is clear that this is an ultrametric distance in the sense just defined. It is also clear that infinity of ultrametric distances can be defined so as to be consistent with the order imposed by the chain CH, and we must remember that the same hierarchy can be specified by several different such chains. Conversely, an **indexed hierarchy** can be considered, given an ultrametric distance.

3 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.

3.1 Confusion in using r instead of s, for a hierarchy H

If r, s ⊂ H, then the **confusion** in using r instead of s, written conf(r, s), is: **(1)** conf (r, r) = conf (r, s) = 0, where s is any ascendant of r; **(2)** conf (r, s) = 1 + conf (r, father_of(s)) ↓ To measure conf, count the *descending* links from r to s, the replaced value. conf is not a *distance*, nor *ultradistance*. To differentiate, we prefer to use **confusion** instead of other linguistic terms like relatedness or closeness.

Example (Hierarchy H_2): conf(r, s) for H_2 of Figure 2 is given in Table 1:

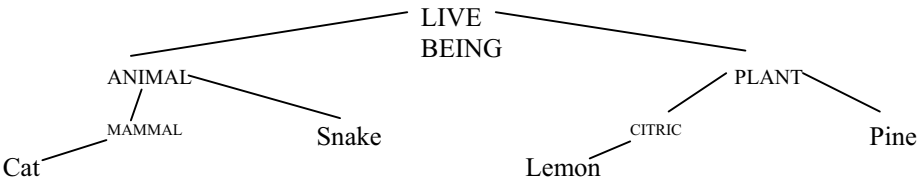

Fig. 2. A hierarchy H_2 of live beings.

Table 1. conf(r, s): Confusion in using r instead of s for the live beings of H_2.

r \ 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

The confusion thus introduced *resembles reality* and *catches the hierarchy semantics*. For example, conf (*animal, 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 animal has error 1; conf (*live_being, animal*) = 1. The confusion among two brothers (say, dog and cat) is 1; using a son instead of the father produces conf=0; using the father instead of the

son makes conf = 1. conf is *not* a symmetric property. Using *general things* (see row 'live being') instead of *specific things* produces *high errors*. Using *specific things* (see row 'lemon') instead of *general things* produces *low errors*. The table's lower triangular half has *smaller errors* than its upper triangular half[5].

3.1.1 Confusion in using r instead of s, for hierarchies that are bags

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

For bags, the **similarity** in using r instead of s, sim^b (r, s), is: **(1)** sim^b (r, r) = sim^b (r, any ascendant_of (r)) = 1; **(2)** if s = some son_of(r), sim^b (r, s) = number of elements of E ~ r ~ s / number of elements of E ~ r = relative popularity of s in r[6]; **(3)** sim^b (r, s) = sim^b (r, some son_of(r)) * sim^b (that son_of(r)), s). ↓

The **confusion** in using r instead of s, conf'(r, s), is 1 – sim^b (r, s). ↓

Example: If *baseball_player* = {*pitcher catcher base_player* ∇ {*baseman baseman baseman*} *field_player* ∇ {*fielder fielder fielder*} *shortstop*} then (a) conf' (*fielder, baseball_player*) = 1 – sim^b (*fielder, baseball_player*) = 0; (b) conf' (*baseball_player, fielder*) = 1 – 1/3 = 2/3; (c) conf' (*baseball_player, left_fielder*) = 8/9 (a *left_fielder* is one of those three fielders); (d) conf' (*base_player, fielder*) = 2/3.

3.1.2 Confusion in using r instead of s, for hierarchies that are lists

For hierarchies that are lists (ordered sets, for instance Temp = {icy, cold, normal, warm, hot, burning}), the **confusion** in using r instead of s, conf'' (r, s), is defined as follows: **(1)** conf'' (r, r) = conf (r, any ascendant of r) = 0; **(2)** If r and s are distinct brothers, conf'' (r, s) = 1 if the father is not an ordered set; else, conf'' (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)** conf'' (r, s) = 1 + conf''(r, father_of(s)). ↓ 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 Ω 1. *Example*: in the list *Temp*, conf'' (*icy, cold*) = 1/5, while conf'' (*icy, burning*) = 5/5.

The rest of the paper will derive results for conf; those for conf' and conf'' can be similarly derived.

3.2 The set of values that are equal to another, up to a given confusion

A **value u is equal to value v, within a given confusion** κ, written $u =_\kappa v$, iff conf(u, v) Ωκ (It means that value u can be used instead of v, within error κ). ↓ *Example*: If v = *lemon* (Figure 2), then (a) the set of values equal to v with confusion 0 is {*lemon*}; (b) the set of values equal to v with confusion 1 is {*citric lemon*}; (c) the set of values

[5] These triangular parts would result to be equal for ultrametric distance. Thus, ultrametrics represents a context-looseness measure in this case.

[6] 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 of s in r.

equal to v with confusion 2 is {*plant citric pine lemon*}. Notice that $=_\kappa$ is neither *symmetric* nor *transitive*.

3.2.1 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 variables,[7] by defining the set S of objects that satisfy predicate P within a given confusion κ.

P holds for object o with confusion κ, or P holds for o within κ, iff (1) if P is formed by non-hierarchical variables, iff P is true for o; (2) for pr a hierarchical variable and P of the form (pr = c), iff for value v of property pr in object o, $v =_\kappa c$ (if the value v of the object can be used instead of c with confusion κ); (3) if P is of the form P1 ⟩ P2, iff P1 holds for o within κ or P2 holds for o within κ; (4) if P is of the form P1 ⟨ P2, iff P1 holds for o within κ and P2 holds for o within κ; (5) if P is of the form ⌡ P1, iff P1 does not hold for o within κ. ↓

Example 1 (refer to hierarchies H_1 and H_2 above): Let the *predicates* be: P = (*lives_in = USA*) ⟩ (*pet = cat*), Q = (*lives_in = USA*) ⟨ (*pet = cat*), R = ⌡ (*lives_in = Spanish_Speaking_Island*); and the *objects* be (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):

Table 2. How the predicates P, Q and R of example 1 hold for several objects.

	P holds within κ for:	**Q holds within κ for:**	**R holds within κ for:**
κ = 0	Ann, Fred, Tom	Fred	Ann, Bill, Fred, Tom
κ = 1	Ann, Fred, Tom	Fred, Tom	Ann, Fred, Tom
κ = 2	Ann, Fred, Tom, Sam	Ann, Fred, Tom	Nobody

3.2.2 The smallest κ for which P(o) is true

How close is Tom to be like Ann in Example 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 = (*lives_in = USA*) ⟨ (*pet = snake*) to Tom, we see that S starts holding for κ=1. The answer to "How close is Tom to Ann?" is 1. Notice that this is not a *symmetric* property.

Ann is close to Tom starting from κ=2; that is, (*lives_in = Mexico*) ⟨ (*pet = cat*) does not hold for Ann at κ=1, but it starts holding for her at κ=2. This defines the "*closeness to*".

Object o **κ-fulfills** predicate P at threshold κ, if κ is the smallest number for which P holds for o within κ. ↓ Such smallest κ is the **closeness** of o to P. ↓ It is an integer number defined between an object and a predicate. The closer is κ to 0, the "tighter" P holds. Compare with the *membership function* for fuzzy sets.

---

[7] For variables that are not hierarchical, a match in value means conf = 0; a mismatch means conf = ←

3.3 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 (Cf. §1.1). Now, consider the case where the *variables* (or relations) form a hierarchy. For instance, relative and brother, in a universe of kinship relations E = {*sister, aunt…*}. Consider *hierarchies H_3 and H_4*: (H_3) *relative* ∇ {*close_relative* ∇ {*father mother son daughter brother sister*} *mid_relative* ∇ {*aunt uncle niece cousin*} *far_relative* ∇ {*grandfather grandmother grandson granddaughter grandaunt granduncle grandcousin grandniece*} }, (H_4) *player* ∇ {*socker_player* ∇ {*John Ed*} *basketball_player* ∇ {*Susan Fred*} }.

In hierarchy H_3, conf (*son, relative*) = 0; conf (*relative, son*) = 2. We know that, for object (Kim (*close_relative Ed*) (*pet cat*)), the predicate V = (*close_relative Ed*) holds with confusion 0. It is reasonable to assume that W = (*son Ed*) holds for Kim with confusion 1;[8] that X = (*relative Ed*) holds for Kim with confusion 0. Moreover, since Ed is a member of hierarchy H_4, it is reasonable to assume that for object (Carl (*close_relative socker_player*) (*pet pine*)) the predicate V holds with confusion 1, X holds with confusion 1 and W holds with confusion 1+1 = 2. Thus, we can extend the definition to variables that are members of a hierarchy, by adding another bullet to the definition of §3.2.1, thus:

If P is of the form (var = c), for var a variable member of a hierarchy, iff) variable var_2 for which (var_2=c) holds for o within κ – conf (var, var_2), where var_2 also belongs to the hierarchy of var.↓ The confusion of the variables *adds* to the confusion of the values. *Example*: For (Burt (*relative basketball_player*) (*pet cat*)), V holds with confusion 1+2=3, W with confusion 2+2=4, and X with confusion 0+2=2.

3.4 Similarity for values in different hierarchies and in different ontologies

When v_1 belongs to a hierarchy H_1 and v_2 to another hierarchy H_2, both with the same element set E, it is best to construct an *ontology* O_U from E, and then to use it to measure the similarity sim'(v_1, v_2), as follows: sim' (c_U, d_U) for two concepts belonging to the *same ontology* O_U, is defined as the 1/(1 + length of the path going from c_U to d_U in the O_U tree). ↓ sim' is defined for *concepts,* not for symbolic values.

Also, for concepts c_A, d_B belonging to *different ontologies* O_A, O_B, we define: sim'' (c_A, d_B) when d_B is *not* the most similar concept in O_B to c_A ⊂ O_A, is equal to s_1s_2, where s_1 = sim (c_A, O_A, O_B) [sim gives the similarity between c_A and its most similar concept c_B in O_B; sim also finds c_B], and s_2 = sim' (c_B, d_B). ↓

3.5 Comments and summing-up

It is worth pausing at this point to look again at ideas of similarity, dissimilarity (confusion) and distance as they apply to a set E. It is difficult in practice to set up a

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

partial order if the number of elements x, y, z... of E is large, and if it is possible it is difficult to make this order without running the risk of generating contradictions. In fact, *the only practical way to establish a partial order is to define a numerical function of similarity or dissimilarity (confusion)* that can be computed in terms of the attributes of every element of E: the **similarity** μ(x,y) will be greater the more closely x resembles y; the **dissimilarity (confusion)** ∂(x,y) will be smaller the more closely x resembles y. The same partial order can be generated by any of an unlimited number such functions. Some dissimilarity functions, however, may not be distances (Cf. §§3.1-3.4). However, we can make simple transformations of ∂(x,y) without affecting the corresponding partial ordering Ψ2, 13βand loss the context. Our point is that it is *not necessary* to do (more arguments in [7]).

Summing-up the analysis presented in previous sections, we can emphasize:
1) Attempting to define a distance on hierarchies of symbolic values to measure closeness between hierarchical elements and hold its partial (total) order, we can lose the *context* of a problem under consideration (§3.1, Table 1).
2) When the context (semantics) of a problem is considered, by expressing the similarity function in terms of the data attributes, we can overcome it (§3.1 and [7] [9]).
3) Such approach finds the set of values that are equal to another up to a given confusion (§3.2) as well as the closeness of an object to the predicate. Similarity functions for values in different hierarchies (or ontologies) can be defined (§3.4 and Ψβ).
4) Hierarchies are simpler than ontologies, although very useful. They are easier to understand, and the extensions to searches, queries and imperfect answers are straightforward (§3.2-3.3 and Ψβ). 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 an ontology, but with *confusion* properties, like a hierarchy [14].

4 Some Applications to Linguistic Analysis[9]

Quasihierarchies and recursive structures have been used in [1] for linguistic analysis of Russian and English texts, verses translation, and computer program comments (fogware). Clasitex [2] is a program that tells us 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.

Recent computational linguistics researches can be linked to our topic as follows.

Information in mostly used WordNet is organized around logical groupings called synsets. Each synset consists of 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*. Two kinds of relations are represented by

[9] We limited these to WordNet due to the page limit. More applications and examples in NLP and several other areas of AI can be found in [7] [9] [15].

pointers: lexical and semantic. Lexical relations hold between word forms; semantic relations hold between word meanings. These relations include (but are not limited to) hypernymy/hyponymy, antonymy, entailment, and meronymy/holonymy. Nouns and verbs are organized into *hierarchies* based on the hypernymy/hyponymy relation between synsets. Additional pointers are used to indicate other relations [5].

Five different proposed measures of similarity or semantic distance in WordNet were experimentally compared by examining their performance in a real-word spelling correction system [8]. It was found that Jiang and Conrath's measure gave the best results overall. That of Hirst-St-Onge seriously over-related, that of Resnik seriously under-related [10], and those of Lin and of Leacock-Chodorow fell in between.

Note that all the measures except of Hirst and St-Onge are *similarity* (not relatedness) measures considering only *the hyponymy hierarchy* of WordNet.

Thus, the measures herein proposed can be compared for at least that hierarchy (§3). Moreover, we shall attempt to compare Hirst-St-Onge's measure and the measure of §3.4 on overall WordNet structure, maybe, by using the same methodology as in [8]. Other issue that can be addressed by our approach is the possibility provided by the definitions of §3.2 for another evaluation method besides those in [8]. Yet other issue is a search for explanation of difference in performance of the "looking arithmetically identical" Jiang-Conrath's and Lin's measures [8]. The prompt is that both measures should be seriously embedded into WordNet context by the interaction procedure of [11]. Our future research will be concerned with these issues. These issues will also be addressed in the now-developing project "Precision-controlled retrieval of qualitative information." We also invite the CL community to test our measures in existing linguistic data bases thus providing some sort of validation.

5 Conclusion

The notions of hierarchy and hierarchical variable make it possible to measure the *confusion* when a value is used instead of another. This makes a natural generalization for predicates and queries. The notions were introduced and developed for arbitrary hierarchies formed by sets, but they can be extended to bags and lists too.

The concepts given herein have practical applications, since they mimic the manner in which people process qualitative values and disambiguate senses (an interesting procedure is [16]). Some examples are given.

References

1. Alexandrov, V., Arsentieva, A.: *Dialogue Structure (Dialogue – Is It an Art or Science?).* Leningrad Inst. for Informatics and Aut. of the USSR Acad. of Sciences (1984).
2. Guzman, A.: Finding the Main Themes in a Spanish Document. *Journal Expert Systems with Applications,* Vol. **14**, No. 1/2 (1998) 139-148
3. Guzman, A., Olivares, J., Demetrio, D., Dominguez, C.: Interaction of Purposeful Agents that use Different Ontologies. *Lecture Notes in Artificial Intelligence*, Vol. **1793**. Springer-Verlag, Berlin Heidelberg New York (2000) 557-573

4. Guzman, A., Dominguez, C., Olivares, J.: Reacting to Unexpected Events and Communicating in spite of Mixed Ontologies. *Lecture Notes in Artificial Intelligence*, Vol. **2313**. Springer-Verlag, Berlin Heidelberg New York (2002) 377-386
5. WordNet: A Lexical Database for the English Language http://www.cogsci.princeton.edu/~wn/
6. Lenat, D.B., Guha, R.V.: *Building Large Knowledge-Based Systems.* Addison-Wesley (1989)
7. Levachkine, S., Guzman, A.: Hierarchies as a New Data Type for Qualitative Variables. Submitted to the *Journal of Data Knowledge Engineering*, Elsevier (2002)
8. Budanitsky, A., Hirst, G.: Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures. Workshop on WordNet and Other Lexical Resources, in the *North American Chapter of the Association for Computational Linguistics* (NAACL-2000), Pittsburgh, PA, June 2001
9. Guzman, A., Levachkine, S.: Graduate Errors in Approximation Queries using Hierarchies and Ordered Sets. Submitted to *MICAI 2004.*
10. Resnik, P.: 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
11. Olivares, J., Guzman, A.: Measuring the Comprehension or Understanding between two Agents (to appear)
12. Simon, J.-C.: *Patterns and Operators. The Foundations of Data Representation*. McGraw-Hill (1984)
13. Alexandrov, V.: *Developing Systems in Science, Technique, Society and Culture.* Nauka, Saint Petersburg (2002)
14. de Gyves, V., Guzman, A.: *BiblioDigital.* ⊕SoftwarePro International (work in progress)
15. Levachkine, S., Guzmaan, A.: Confusion between hierarchies partitioned by a Percentage rule. Submitted to *MICAI 04.*
16. A. Gelbukh. Using a semantic network for lexical and syntactical disambiguation. Proc. CIC-97, *Simposium Internacional de Computación,* 12-14, 1997, CIC, IPN, Mexico City, Mexico, 352–366. www.gelbukh.com/CV/Publications/1997/CIC-97-Sem-Net.htm