Technical Report
N° 1/2006

Semantic Similarity Tailored on Application Context

Riccardo Albertoni,
Alessio Bertone,
Monica De Martino

Istituto di Matematica Applicata e Tecnologie Informatiche,
Consiglio Nazionale delle Ricerche
Via de Marini 6, 16149 Genoa, Italy

Semantic Similarity tailored on the Application Context

Riccardo Albertoni, Alessio Bertone, Monica De Martino

CNR-IMATI,
Via De Marini, 6 – Torre di Francia - 16149 Genova, Italy
{albertoni,bertone, demartino}@ge.imati.cnr.it

Abstract. The paper proposes an approach to assess the semantic similarity among instances of an ontology. It aims to define a sensitive measurement of semantic similarity, which takes into account different hints hidden in the ontology definition and explicitly considers the application context. The similarity measurement is computed by analyzing, combining and extending some of the existing similarity measures and tailoring them according to the criteria induced by specific application context.

1 Introduction

In this decade the ontologies have been imposing in the computer science as artifact to explicitly represent shared conceptualization. A remarkable research effort has been spending to develop new ontology languages, proper reasoning mechanisms and correlated management tools, but less attention is generally posed on the similarity among the ontology instances.

Despite the similarity plays a central role in several activity as information retrieval, exploration and analysis, the most of research activity concerning the similarities has been carried out within the field of ontology alignment. However, as similarities for the ontology alignment strongly focus on the comparison of the structural parts of distinct ontologies, their adoption for assessing the similarity among instances belonging to an ontology might result misleading. A part from the similarity for ontology alignment, few other methods to assess similarities among instances and concepts have been proposed but they are far from being adopted as a standard framework in the similarity assessment. Unfortunately, they ignore that the ontology entities (classes, attributes, relations) might differently concur to define the similarity according to the application context where the similarity is defined. Moreover the ontology, as a formalization of a conceptualization, encodes many implicit information providing hints useful to suitably define the semantic similarity but the existing methods partially consider them.

The paper proposes a more sensitive measurement of semantic similarity considering some hints provided by the ontology and explicitly tailored on the application context. The basic concept in our approach is to assume that the information related to the application context play an important role in the similarity assessment, thus the first mission is to identify and formalise the criteria related to the application context, which affect the similarity measurement. Then the similarity among instances is defined by an amalgamation function, which aggregates different similarities, taking into account the influences of all the ontology

entities. The similarity measurements have been defined analyzing, combining and extending some of the existing similarity measures. Because of the lack of space only similarity with asymmetric property is considered in the paper.

2 Related work

Semantic similarity is an important concept, which seems to be designated to play a prevalent role in different fields of the Semantic Web. Currently it is relevant in the ontology alignment [1,2], conceptual retrieval [3] as well as semantic web service discovery and matching [4,5] and it is expected to increase its relevance in context as the metadata analysis [6].
In the following some works related to the ontology similarity are shortly described:

- Ontology alignment

There are plenty of methods to align ontology, as pointed out by Euzenat et al. [2]. The semantic similarity is adopted in this context to figure out relations among the entities in the ontology schema. It is employed to compare the name of classes, attributes and relations, determining reasonable mapping between the ontologies. Some similarities adopted for the ontology alignment consider also quite expressive ontology language, (e.g., [1] focus on a subset of OWL Lite) but they mainly focus on the comparison of the structural aspects of ontology not on the similarity between instances as intended in this paper.

- Similarity among elements of a lexicographic databases

Different approaches to assess semantics similarity among concepts represented by words within lexicographic databases are available. They mainly rely on edge counting-base [7] or information theory-based methods [8]. The edge counting-base method assumes terms which are subjects of the similarity assessment as edges of a tree-like taxonomy and defines the similarity in terms of the distance between edges [7]. The information theory-based method defines the similarity of two concepts in terms of the maximum information content of the concept which subsumes them [9,10]. Recently new hybrid approaches have been proposed: Rodriguez and Egenhofer [11] takes advantage from the above methods and adds the idea of features matching introduced by Tversky [12]. Schwering [3] proposes a hybrid approach to assess similarity among concepts belonging to a semantic net. The similarity in this case is assessed comparing properties of concept as feature [12] or as geometric space [13].
In general these semantic approaches adopt ontology models that are not standard in the semantic web. On the one hand, Rada et al. [7], Resnik [9], Lin [10] work on lexicographic ontologies where the instances are not considered or are quite different from the instances intended in some language as RDF(S) or OWL. They could be applied to define a similarity among instances but they are doomed to fail since they ignore important information provided by the instances attributes and relations. On the other hand, Rodriguez and Egenhofer [11] and Schwering [3] use the features or even conceptual spaces, information that are not native in the ontology and should be manually added.

- Similarity among elements of an ontology language comparable to semantic web standards

Other works define similarity relying on ontology models closer to those adopted in the semantic web standards. Hau et al. [5] identifies similar services measuring the similarity between their descriptions. To define a similarity measure on semantic services it explicitly refers to the ontology model of OWL Lite and defines the similarity among OWL objects

(classes as well as instances) in terms of the number of common RDF statements that characterize the objects. Maedche and Zacharias [14] adopts a semantic similarity measure to cluster ontology based metadata. The ontology model adopted in this similarity refers also to IS-A hierarchy, attributes, relations and instances. The similarity is worked out considering hierarchies, attributes and relations shared by classes and instances. Even if these methods define similarity relying on ontology models which are more evolved than the taxonomy or terminological ontology, their design generally assume choices which are arguable in a real case study. For example in Hau et al. [5] the statements which are relevant for the similarity assessment are determined by fixing a distance (*degree of description set*) within the statements considered in the neighborhood. In our opinion this distance cannot be assumed independently from the entity consider during the similarity assessment. In the approach proposed by Maedche and Zacharias [14] all attributes or relations of a given class are relevant to determine the similarity among instances while their importance should depend on the context. Due to this simplification the mention similarity measurements fail to provide the tool to tailor the semantic similarity according to specific purposes in different application contexts.

3 Semantic Similarity

In this paper a semantic similarity among instances of an ontology is defined taking advantage from the similarity hints hidden in the ontology definition and by considering explicitly the application context.
To precisely define the similarity, the definitions of the ontology model and the similarity are given. In particular, in this paper the ontology model with data type defined by Ehrig et al.[15] is adopted.

Definition 1: Ontology with Data Type

An Ontology with data type is a structure
$O := (C, T, \leq_c, R, A, s_R, s_A, \leq_R, \leq_A, I, V, l_C, l_T, l_R, l_A)$ *where:*

- *C,T,R,A,I,V are disjointed sets respectively containing classes, data types, relations, attributes, instances and data values,*
- $\leq_C$ *is the partial order on C, which defines the classes hierarchy,*
- $\leq_R$ *is the partial order on R which defines the relation hierarchy,*
- $\leq_A$ *is the partial order on A which defines the attributes hierarchy,*
- $s_R : R \to C \times C$ *is the function that provides the signature for each relation,*
- $s_A : A \to C \times T$ *is the function that provides the signature for each attribute,*
- $l_C : C \to 2^I$ *is the function called class instantiation,*
- $l_T : T \to 2^V$ *is the function called data type instantiation*
- $l_R : R \to 2^{I \times I}$ *is the function called relation instantiation*
- $l_A : A \to 2^{I \times V}$ *is the function called attribute instantiation*

Two kinds of similarity can be adopted: similarity with symmetric or with asymmetric properties.

Definition 2: Normalized Symmetric and Asymmetric Similarities

A symmetric normalized similarity $S : IxI \rightarrow [0,1]$ *is a function that maps a pair of instances to a real number in the range [0,1] such that:*

$\forall x, y \in I \quad S(x, y) \geq 0$ *Positiveness*

$\forall x \in I, \forall y, z \in I, S(x, x) \geq S(y, z)$ *Maximality*

$\forall x, y \in I \quad S(x, y) = S(y, x)$ *Symmetry*

A normalized asymmetric similarity is a function $\overline{S} : IxI \rightarrow [0,1]$ *where the symmetry axiom is not satisfied.*

The preference between symmetric and asymmetric similarity mainly depends on the scenario where the similarity is applied, there is no a-priori reason to formulate this choice. A complete framework to assess the semantic similarity among instances should provide both types of similarity. In this paper only the asymmetric similarity is described due to the lack of space.

The proposed approach adopts the schematisation of the similarity framework defined by Ehrig et. al. [15]. Ehrig et. al. structures the similarity in terms of Data, Ontology and Context layers plus the domain knowledge layer which spans all the other. The data layer measures the similarity of entities by considering the data values of simple or complex data types such as integer and string. The ontology layer considers the similarities induced by the ontology entities and the way they are related each other. The Context layer assesses the similarity according to how the entities of the ontology are used in some external contexts.

The framework defined by Ehrig is suitable to support the ontology similarity as well as instances similarity. In the paper the framework is extended and specialized to define a similarity among instances. Our contribution consists of an accurate formalization of the Context and Ontology layer. Concerning the data and domain knowledge layers the paper adopts a replica of what is illustrated in [15].

In the context layer the formalization to express the similarity criteria induced by the application context is provided. The ontology layer combines and extends some existing methods to work out the similarity, it takes into account the criteria induced by the context as well as the hints scattered all over the ontology definition. The formalization of the application context is employed to parameterise the computation of the similarity in the ontology layer, forcing it to adhere to the application criteria.

The overall similarity is defined by an amalgamation function, which aggregates some similarity functions defined within the ontology layer.

Definition 3: Amalgamation Function

Let be $\overline{Sim}$ *the overall similarity between two instances,* $\overline{ExternSim}$ *and* $\overline{ExtensSim}$ *two similarity functions defined in the ontology layer,* $w_{ExternSim}$ *and* $w_{ExtensSim}$ *the weights to balance the functions importance.* $\overline{Sim}$ *is defined by:*

$$\overline{Sim}(i_1, i_2) = \frac{w_{ExternSim} * \overline{ExternSim}(i_1, i_2) + w_{ExtensSim} * \overline{ExtensSim}(i_1, i_2)}{w_{ExternSim} + w_{ExtensSim}} \quad (1)$$

In the paper, $w_{ExternSim}$ and $w_{ExtensSim}$ are equal to 1\2.

In the below sections the context layer is described as well as the two similarities $\overline{ExternSim}$ *and* $\overline{ExtensSim}$ at the ontology layer.

3.1 Context Layer

Contexts are considered as local models that encode a party's subjective view of a domain. The context layer is defined to assess the similarity according to how the entities are used in some external contexts [15]. This paper focuses on the Application Context, which explains how an entity of an ontology is used in the context of a given application. A formalization of the Application Context is defined providing a language to be adopted to tune the similarity assessment.

3.1.1 Application Context

The Application Context is a formalization of how the context affects the choice of the attributes and relations to be considered in the similarity assessment.

Two factors influence the choice of the attributes and relations of a given ontology:

- the class or the path to reach the class which the attributes and relations belong to,
- the criteria adopted to compare the attributes and relations.

Concerning the first factor, given an Application Context, not all the attributes and relations contained in the class have the same importance in the similarity assessment. Then the formalization of the Application Context should provide hints about the choice of the attributes and relations to be considered. Moreover, since the similarity between instances can be defined both in terms of the classes they belong to and recursively in terms of the classes having related instances, if we suppose to reach the classes by navigating the class relations, the path on the ontology graph induced by the navigation has to be taken into account!

Let consider an example: supposing to have the ontology describing the research departments and to be interested in the similarities among researchers' and publications' instances. The similarity among researchers might be defined considering the common projects, their age, their shared publications and their participation at the same scientific events. On the other hand, the similarity among publications is defined in terms of type of publication (journal, conference proceeding, workshop, book, book chapter), date of publication, topics and the similarity among the co-authors. The comparison of researchers' instances are involved in both the similarity assessments since to assess the similarity between two publications we recursively consider the similarity among the researchers who are the co-author. However not the same attributes are relevant for the comparison: for instance the attribute "age of the researchers" is functional in the assessment of the similarity between researchers and not in the recursively assessment of the similarity among publications.

Concerning the latter factor, three different criteria to compare the attributes and relations are identified:

- Criteria based on the cardinality of the attributes or of the relations: the similarity is assessed according to the number of instances the relations have, or the number of values that an attribute assumes. For example, two researchers can be regarded as similar if they have a similar "number" of publications. We call *Count* the parameter to identify the cardinality.
- Criteria based on the intersection between the set of attributes or of relations: the similarity is assessed according to the number of elements they have in common. For example, the more papers two researchers have in "common", the more they are similar. We call *Inter* the parameter to evaluate the intersection.
- Criteria based on the similarity of attributes and relations: the similarity is assessed in terms of similarity of attributes values and related instances, For example two researchers are as similar as they have "similar" publications. We call *Simil*, the parameter to evaluate the similarity.

Thus to provide an accurate formalism for the Application Context, it is needed to model these two factors.

To be more precise the application context is expected to be provided by an ontology engineer according to specific application needs. The following formalization provides the restrictions that the Application Context must adhere to. In particular, the application context formalization is given relying on the concepts of "sequence of elements belonging to a set X", "path of recursion" and "set of path of recursion".

Definition 4: Sequences of a Set X

Given a set X, a sequence s of elements of X with length n is defined by the function

$$s:[1,..,n]\to X, n\in N^{+} \tag{2}$$

It is represented in simple way by the list [s(1),..,s(n)].

Let be $S_X^n=\{s\mid s:[1,n]\to X\}$ *the set of sequences on X having length n and* $\cdot: S_X^n x S_Y^m \to S_{X\cup Y}^{n+m}$ *the operator "concat" between two sequences.*

Definition 5: Path of Recursion

A path of recursion p with length i is a sequences that satisfies the follow conditions

$$p\in S_{C\cup R}^{i}\wedge p(1)\in C\wedge\forall j\in[2,i]\ p(j)\in R \tag{3}$$

Let name P the set of paths of recursion defined by,

$$P=\bigcup_{i\in N^{+}}P^{i}=\bigcup_{i\in N^{+}}\{p\in S_{C\cup R}^{i}\mid p(1)\in C\wedge\forall j\in[2,i]\ p(j)\in R\} \tag{4}$$

Moreover, let define the following functions in terms of the ontology model with data type:

- $d_a: C\to 2^{A}; d_a(c)=\{a:A\mid \exists\, t\in T, s_A(a)=(c,t)\}$ the set of attributes of $c\in$ C.
- $d_a: R\to 2^{A}; d_a(r)=\{a:A\mid \exists c,c'\in C\ \exists\, t\in T\ s_R(r)=(c,c')\wedge s_A(a)=(c',t)\}$ the set of attributes of r∈ R.
- $d_r: C\to 2^{R}; d_r(c)=\{r:R\mid \exists\, c'\in C, s_R(r)=(c,c')\}$ the set of relations of $c\in$ C.

- $d_c : R \to 2^C; d_c(r) = \{c' : C \mid \exists\, c \in C \;\; s_R(r) = (c, c')\}$ the set of concepts reachable through r.
- $d_r : R \to 2^R; d_r(r) = \{r' : R \mid \exists\, c \in C, \exists\, c' \in d_C(r); s_R(r') = (c', c)\}$ the set of relations of the concepts reachable through r .
- $d_c : C \to 2^C; d_c(c) = \{c' : C \mid \exists\, r \in d_r(c) \;\; s_R(r) = (c, c')\}$ the set of concepts related to $c \in$ C through a relation.

Definition 6: Application Context

Given the set P set of paths of recursion, let call $L = \{Count, Inter, Simil\}$ *the set of criteria adopted to compare the attributes and relations of a class, the Application Context is defined by a function AppCont adhering to the follow signature.*

$$AppCont : P \to (2^{A \times L}) \times (2^{R \times L}) \qquad (5)$$

The signature of AppCont is defined more precisely in inductive way on the length of the path of recursion where:

1. *Base (AppCont for path of recursion with length equal to 1)*

$$p \in P^1 \xrightarrow{AppCont} (attr, rel) \in (2^{d_a(p(1)) \times L}) \times (2^{d_r(p(1)) \times L}) \qquad (6)$$

2. *Inductive step: starting from the AppCont on a path of recursion having length n*

$$p \in P^n \xrightarrow{AppCont} (attr, rel) \in (2^{d_a(p(n)) \times L}) \times (2^{d_r(p(n)) \times L}) \quad p(n) \in R \qquad (7)$$

the AppCont on a path of recursion having length equal to n +1 is defined by:

$$p' \in P^{n+1} \xrightarrow{AppCont} (attr, rel) \in (2^{d_a(p'(n+1)) \times L}) \times (2^{d_r(p'(n+1)) \times L}) \quad \begin{array}{l} r \in S_R^1; p \in P^n; p' = p \cdot r \\ AppCont(p) = (attr, rel) \\ \wedge\, (r, Simil) \in rel \end{array} \qquad (8)$$

In particular, let X be an instantiation of 'R' or 'A', $AppCont_A$ and $AppCont_R$, two operators which define two functions $AppCont_R : P \to 2^{R \times L}$ *and* $AppCont_A : P \to 2^{A \times L}$ *so that for each path of recursion they return respectively the set of relevant relations and the set of relevant attributes, and satisfy the condition of univocity in the recursion path .*

$$\begin{array}{l} \forall p \in P, \forall (x, l) \in 2^{(A \cup R) \times L}\ \exists (x', l') \in 2^{(A \cup R) \times L} \\ (x, l) \in AppCont_X(p) \wedge (x', l') \in AppCont_X(p) \Leftrightarrow (x, l) = (x', l') \end{array} \qquad (9)$$

3.2 Ontology layer

In the ontology layer the similarity functions which compose the amalgamation function (definition 3) are defined. In particular, two similarities with asymmetric properties are defined respectively according to a "structural comparison" or an "extensional comparison". The "structural comparison" measures the instances similarity at the level of ontology

schema: given two instances, it compares the classes they belong to considering the attributes and relations shared by the classes and their position within the class hierarchy. The "extensional comparison" compares the extension of the ontology entities; in practice, it bases the similarity assessment on the attributes values as well as the related instances.
In the ontology layer additional hypotheses concerning the ontology model and the similarity are introduced:

- All classes defined in the ontology have the fake class *Thing* as super-class.
- Given $i_1 \in l_c(c_1)$, $i_2 \in l_c(c_2)$, if c_1 c_2 have a common super-class different from *Thing*, their similarity is equal to 0.
- The least upper bound *(lub) between* c_1 c_2, which is defined as the immediate super-class of c_1 c_2 that subsumes both classes, is unique.

This hypothesis at first sight might appear quite strong but it could be clearly motivated considering the role played by the IS-A relation. Additionally, they aim to force the *lub* to be a sort of "template class" providing the attributes and relations shared by the instances and through which it is possible to perform the instances comparison.

3.2.1 Similarity according to structural comparison

The similarity function $\overline{ExternSim}$ performs the structural comparison between two instances $i_1 \hat{I} l_c(c_1)$, $i_2 \hat{I} l_c(c_2)$ in terms of the classes c_1, c_2 that the instances belong to. More formally $\overline{ExternSim}(i_1, i_2) = \overline{ExternSim}(c_1, c_2)$ with $i_1 \in l_c(c_1), i_2 \in l_c(c_2)$.

Definition 7: ExternSim similarity

Let define two similarities Slots Matching($\overline{SM}$) and Classes Matching($\overline{CM}$) and the respectively weights w_{SM} and w_{CM} in the range [0,1]. The similarity between two classes according to the external comparison is defined by:

$$\overline{ExternSim}(c_1, c_2) = \begin{cases} 1 & if \quad c_1 = c_2 \\ \dfrac{w_{SM} * \overline{SM}(c_1, c_2) + w_{CM} * \overline{CM}(c_1, c_2)}{w_{SM} + w_{CM}} & Otherwise \end{cases} \qquad \textbf{(10)}$$

w_{SM} and w_{CM} are defined for the purpose of this paper equal to 1\2.
The Classes Matching similarity is characterized by the distance between two classes with respect to the hierarchies induced by $\leq_C$ as well as the depth in the hierarchy where the classes lay. On the contrary the Slots Matching similarity is based on the shared attributes and relations. Moreover it is affected by the number of attributes and relations shared by the two instances as well as the overall number of attributes and relations of the instances. The rationale behind it is that two classes having a plenty of attributes and only few attributes in common are less similar than two classes having less attributes but the same attributes in common.
The two similarities are defined in the following paragraphs.

3.2.1.1 Classes Matching Similarity ($\overline{CM}$)

Classes Matching is a similarity evaluated in terms of distance of the classes with respect to the IS-A hierarchy. In this paper it is based on the concept of Upwards Cotopy (*UC)* of a

class c_i. UC represents the set of classes containing c_i and all classes having it as subclasses. We considered the definition of UC provided by [14] and adapting it to an asymmetric similarity.

Definition 8: Upwards Cotopy (UC)

The Upwards Cotopy with respect to a set X and an associated partial order $\leq_x$ is:

$$UC_{\leq_X}(x_i) = \{x_j \in C \mid (x_i \leq_X x_j) \vee x_i = x_j\} \quad \textbf{(11)}$$

The Upwards Cotopy with respect to the set of classes C and an associated partial order $\leq_C$ is defined by: $UC_{\leq_C}(c_i) := \{c_j \in C \mid (c_i \leq_C c_j) \vee c_i = c_j\}$.

$UC_{\leq_C}(c_i)$ can be thought as the set of classes composing the path to reach the furthest super-class (*Thing*) of the hierarchy from c_i.

Definition 9: ***Class Matching Asymmetric Similarity***
Given two classes c_1, c_2,, the Upwards Cotopy of the set C and an associated partial order $\leq_C$, the Class Match similarity with asymmetric property is defined by:

$$\overline{CM}(c_1, c_2) := \frac{\left|UC_{\leq_C}(c_1) \cap UC_{\leq_C}(c_2)\right|}{\left|UC_{\leq_C}(c_1)\right|} \quad \textbf{(12)}$$

$\overline{CM}$ Similarity captures the similarity between two classes considering the number of classes that are in common in the hierarchy. Of course the relevance of the common classes increases of importance with the decreasing of classes needed to join c_1 to the root of the hierarchy.

3.2.1.2 Slot Matching Similarity
Concerning the similarity with respect to shared slots , the one proposed by Rodriguez and Egenhofer [11] can be borrowed . It is based on the concept of distinguishing features which are employed to differentiate subclasses from their super-class. In their proposal, different kinds of distinguishing features are considered (i.e. attributes, functionalities, and parts) but no one coincides immediately with the native entities in the ontology model. The aims of our approach are to assess similarity among classes within a well defined ontology (see definition 1). Of course it would be possible to manually annotate the classes adding the distinguishing features but our approach prefers to focus on what is already available in the ontology model. Therefore attributes and relations are mapped as a kind of distinguishing features. The asymmetric similarity coherently defined by Rodriguez and Egenhofer, is extended to the ontology model taking into account the partial order on the relations ($\leq_R$) and the attributes ($\leq_A$).

Definition 10: Slot Matching Similarity

Given two classes c_1,c_2, two kinds of features (attributes and relations), w_a, w_r, the weights of the features, the similarity function $\overline{SM}$ between c_1 and c_2 is defined in terms of the weighted

sum of the similarities $\overline{S}_a$ *and* $\overline{S}_r$, *where* $\overline{S}_a$ *is the slot matching according to the attributes and* $\overline{S}_r$ *in the slot matching according to the relations.*

$$\overline{SM}(c_1, c_2) = w_a \cdot \overline{S}_a(c_1, c_2) + w_r \cdot \overline{S}_r(c_1, c_2) \quad \textbf{(13)}$$

The sum of weights is expected to be equal to 1, and for simplicity we assume to be equal, therefore $w_a=w_r=1/2$.

The two slots matching ($\overline{S}_a$) and ($\overline{S}_r$) rely on the definitions of *slot importance* and *slot similarity* as defined in the following.

Definition 11: Function a **of "Slot Importance"**

Let c_1, c_2, *be two distinct classes, d the class distance in term of edges in a IS-A hierarchy and lub the immediate super-class that subsumes both classes.* a *is the function that evaluates the importance of the difference between the two classes.*

$$a(c_1, c_2) = \begin{cases} \dfrac{d(c_1, \text{lub})}{d(c_1, c_2)} & d(c_1, \text{lub}) \le d(c_2, \text{lub}) \\ 1 - \dfrac{d(c_1, \text{lub})}{d(c_1, c_2)} & d(c_1, \text{lub}) > d(c_2, \text{lub}) \end{cases} \quad \textbf{(14)}$$

The value of a is between 0 and 0.5. In particular, $\alpha=0$ if the differences of a class with respect to the other are the only important differences for the evaluation of the similarity, $\alpha=0.5$ if the differences of both classes are equally important.

Definition 12: Slots Similarity

Let t be a kind of distinguishing feature (t= attribute or t= relation), X and Y sets of elements of a kind of distinguishing feature t, $x \hat{I} X$, $y \hat{I} Y$ *two slots.*
The similarity between two slots $x \hat{I} X$, $y \hat{I} Y$ *is defined by:*

$$Sim_{\le_t}(x, y) = \frac{|UC_{\le t}(x) \cap UC_{\le t}(y)|}{|UC_{\le t}(x) \cup UC_{\le t}(y)|} \quad \textbf{(15)}$$

The slot similarity between two sets X and Y of elements of a kind of distinguishing feature t with respect to the related hierarchy $\le_t$ *is defined by:*

$$Sim_{\le_t}(X, Y) = \frac{\max_{f \in \{g: X \to Y \mid g \text{ bijective}\}} \left(\sum_{x \in X} Sim_{\le_t}(x, f(x)) \right)}{Min(|X|, |Y|)} \quad \textbf{(16)}$$

Definition 13: Slot Matching asymmetric similarity according to the feature t

Given two classes c_1 *(target) and* c_2, *(base), let be:*

- C_1^t *and* C_2^t *the sets of features of type t respectively of* c_1 *and* c_2;
- $\tilde{C}_1^t = C_1^t \setminus (C_1^t \cap C_2^t)$ *and* $\tilde{C}_2^t = C_2^t \setminus (C_1^t \cap C_2^t)$ *respectively the set of distinguishing features that* C_1^t *does not share with* C_2^t *and* C_2^t *does not share with* C_1^t;

- $\cap_t$ *the intersection among sets of features of type t according to hierarchy* $\leq_t$ *defined by* $\left|C_1^t \cap_t C_2^t\right| = \left|C_1^t \cap C_2^t\right| + Sim_{\leq_t}(\tilde{C}_1^t, \tilde{C}_2^t)$;
- $\setminus_t$ *the sets difference according to hierarchy* $\leq_t$ *defined by* $\left|C_1^t \setminus_t C_2^t\right| = \left|C_1^t \setminus C_2^t\right| - Sim_{\leq_t}(\tilde{C}_1^t, \tilde{C}_2^t)$;

The Slot Matching similarity $\overline{S}_t(c_1, c_2)$ *according to the feature t with asymmetric property is defined by:*

$$\overline{S}_t(c_1, c_2) = \frac{\left|C_1^t \cap_t C_2^t\right|}{\left|C_1^t \cap_t C_2^t\right| + a(c_1, c_2)\left|C_1^t \setminus_t C_2^t\right| + (1 - a(c_1, c_2))\left|C_2^t \setminus_t C_1^t\right|} \quad \textbf{(17)}$$

3.2.2 Similarity according to the Extensional Comparison

The extension of entities plays a fundamental aspect in the assessment of the similarity among the instances. Supposing to assess the similarity of two instance i_1,i_2, it is possible to determine their classes, and to consider their lub. The lub provides a common base to compare the instances, belonging to different classes, since it represents instances of attributes and relations, which are expected to be in common. Finally, the comparison of instances with respect to the lub and the Application Context provides information about what attributes and relations must be considered. The similarity by extensional comparison is characterised by two similarities: a similarity comparing the attributes of the instances and a similarity comparing the relations of the instances.

Definition 14: Extensional Asymmetric Similarity

Given two instances $i_1 \hat{I} l_c(c_1)$, $i_2 \hat{I} l_c(c_2)$, $c=lub(c_1,c_2)$, $p \hat{I} P$ *by p=[c] a path of recursion. Let* $\overline{Sim}_a^p(i_1,i_2)$ *and* $\overline{Sim}_r^p(i_1,i_2)$ *be the similarity measurements between instances considering respectively the attributes and the relations. The extensional similarity with asymmetric property is defined:*

$$\overline{ExtensSim}(i_1, i_2) = \begin{cases} 1 & i_1 = i_2 \\ \overline{Sim}_I^p(i_1, i_2) & Otherwise \end{cases} \quad \textbf{(18)}$$

Where $\overline{Sim}_I^p(i_1, i_2)$ *is the overall similarity between instances of the set I defined by:*

$$\overline{Sim}_I^p(i_1, i_2) = \frac{\sum_{a \in d_a(c)} \overline{Sim}_a^p(i_1, i_2) + \sum_{r \in d_r(c)} \overline{Sim}_r^p(i_1, i_2)}{|d_a(c)| + |d_r(c)|} \text{ where } p \in P \quad \textbf{(19)}$$

The index p is a kind of stack of recursion adopted to track the navigation of relations whenever the similarity among instances is defined in terms of related instances. $\overline{Sim}_a^p(i_1,i_2)$ and $\overline{Sim}_r^p(i_1,i_2)$ are defined by a unique equation in the following definition.

Definition 15: Similarity on Attributes and Relations
Given two instances $i_1 \hat{I} l_c(c_1)$, $i_2 \hat{I} l_c(c_2)$, $c=lub(c_1,c_2)$, $p \hat{I} P$ by $p=[c]$ a path of recursion, let be:

? $i_A(i) = \{v \in V \mid (i,v) \in l_A(a), \exists x \in C \; s.t. \; s_A(a) = (x,T) \wedge l_T(T) = 2^V\}$ *the set of values assumed by the instance i for the attribute a,*

? $i_R(i) = \{i' \in l_c(c') \mid \exists c \; i \in l_c(c) \exists c' \; s.t. \; s_R(r) \in (c,c') \wedge (i,i') \in l_R(r)\}$ *the set of instances related to the instance i by the relation r,*

? *AppCont the Application Context defined according to the restriction in section 3.1*

? $F_X = \{g : i_X(i_1) \to i_X(i_2) \mid g \text{ is bijective}\}$.

The similarity between instances according to their attributes or relations is.

$$\overline{\mathrm{Sim}}_x^p(i_1,i_2) = \begin{cases} 0 & \begin{array}{l} if\ ((x,Simil) \in AppCont_X(p) \\ \wedge \\ (i_X(i_1) \vee i_X(i_2) \text{ are empty sets}) \end{array} \\ 0 & If\ \neg(\exists l \in L \; s.t. \; (r,l) \in AppCont_R(p)) \\ \dfrac{|i_X(i_1)|}{\max(|i_X(i_1)|,|i_X(i_2)|)} & if\ (x,Count) \in AppCont_X(p) \\ \dfrac{|i_X(i_1) \cap i_X(i_2)|}{|i_X(i_1)|} & if\ (x,Inter) \in AppCont_X(p) \\ \dfrac{\max\limits_{f \in F} \sum\limits_{v \in i_A(i_1)} \overline{Sim}_T^a(v, f(v))}{\min(|i_A(i_1)|,|i_A(i_2)|)} & if\ (x=a) \wedge (a,Simil) \in AppCont_A(p) \\ \dfrac{\max\limits_{f \in F} \sum\limits_{i \in i_R(i_1);} \overline{Sim}_I^{pNew}(i, f(i))}{\min(|i_R(i_1)|,|i_R(i_2)|)} & \begin{array}{l} if\ (x=r) \wedge (r,Simil) \in AppCont_R(p) \\ pNew = p \cdot s,\ s \in S_R^1, s(1) = r \end{array} \end{cases} \quad \textbf{(20)}$$

$\overline{Sim}_T^a$ is the similarity defined for the attribute a having data type T. It will be provided by the data layer as suggested by [15]. It is important to note that each time the similarity is assessed in terms of related instances (that is $(r,Simil) \in AppCont_R(p)$) the relation that is followed to reach the related instances is added to track of recursion. Thus it is possible to apply the proper AppCont to the correct path of recursion.

4 Application Example

Let consider part of the ontology KA[1] which defines concepts from academic research (**Fig. 1**) and let focus on two distinct applications: a comparison of the researchers according to their experiences and a comparison of the researchers with respect to their research interests. These analyses are performed evaluating the similarity with respect to two distinct Application Contexts: let call $AppCont_{Exp}$ and $AppCont_{Inte}$ the application context respectively for the research experience analysis and for the research interest analysis. They are defined in the formulas 21 and 22. The similarity among researchers with respect to the $AppCont_{Exp}$ is defined considering the number of publications and projects they have. Roughly, two researchers are assessed as similar if they have a similar number of publications and projects. Considering the second Application Context $AppCont_{Inte}$ two researchers are assessed as similar if they have publications and projects that are common and similar research interests. The attributes of the ontology are not considered in both the Application Contexts. This is a simple example which aims to point out how the formalization of Application Context provides a mean to tailor the similarity measure according to the Application Context, and to demonstrate that also starting from a unique ontology the similarity has to be tailored according to the Application Context.
As final remark, it is worth to note that both the mentioned contexts result in terminating similarity assessments. In some sense, this happens since the functions representing the Application Contexts are numerable and finite.

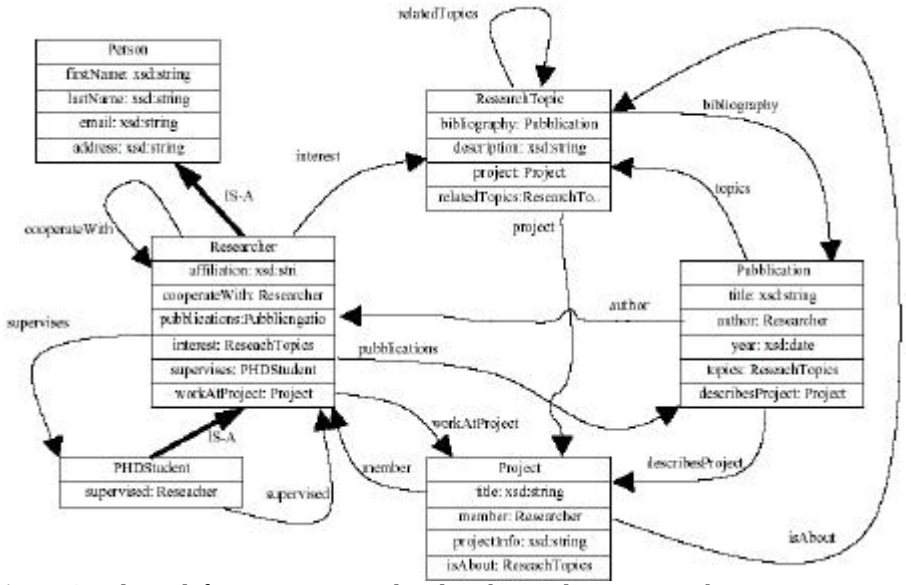

Fig. 1. Ontology defining concepts related to the academic research.

[1] http://protege.stanford.edu/plugins/owl/owl-library/ka.owl

$$[\text{Researcher}] \xrightarrow{AppCont_{Exp}} \{\{f\}, \{(\text{publications}, \text{Count}), (\text{workAtProject}, \text{Count})\} \quad (21)$$
$$[\text{PhdStudent}] \xrightarrow{AppCont_{Exp}} \{\{f\}, \{(\text{publications}, \text{Count}), (\text{workAtProject}, \text{Count})\}$$

$$[\text{Researcher}] \xrightarrow{AppCont_{Inte}} \{\{f\}, \{(\text{publications}, \text{Inter}), (\text{workAtProject}, \text{Inter}), (\text{interest}, \text{Simil})\} \quad (22)$$
$$[\text{Researcher}, \text{Interest}] \xrightarrow{AppCont_{Inte}} \{\{f\}, \{(\text{relatedTopics}, \text{Inter})\}\}$$
$$[\text{PhdStudent}] \xrightarrow{AppCont_{Inte}} \{\{f\}, \{(\text{publications}, \text{Inter}), (\text{workAtProject}, \text{Inter}), (\text{interest}, \text{Simil})\}$$
$$[\text{PhdStudent}, \text{Interest}] \xrightarrow{AppCont_{Inte}} \{\{f\}, \{(\text{relatedTopics}, \text{Inter})\}\}$$

5 Conclusion and Future Work

The paper proposes an approach to assess the semantic similarity given a precise ontology model. It combines and extends different existing similarity methods taking into account the hints scattered both in the external and extensional part of the ontologies. The formalization of different Application Contexts is provided as a mean to parameterise the similarity assessment, and to formulate a measurement more sensible to the specific application needs. Since similarity is expected to play a crucial role in the Semantic Web, we believe that our approach will become an important tool to support the analysis task.

Nevertheless some research and development issues are still open. For example in the paper only the asymmetric similarity has been defined, there is not an a-priori reason. We are aware that a complete framework requires both symmetric and asymmetric similarity according to the scenario where it has to be applied. Moreover in the proposed approach the Application Context affects only the similarity defined by the extensional comparison. It could be interesting to further analyse if the context result also in the external comparison similarity. Finally, it would be worth to precisely formalize the conditions related the path of recursion to ensure the termination of the similarity assessment, to extend the similarity to ontology model towards OWL and to test it on a specific user case.

6 Acknowledgements

This research started within the EU founded INVISIP project and then has been partially performed within the Network of Excellence AIM@SHAPE.

7 References

1. Euzenat, J. and Valtchev, P.: Similarity-Based Ontology Alignment in OWL-Lite. ECAI. (2004) 333-337
2. Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., De Bo, J., Dieng, R., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Maynard, D., Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaiko, P., Tessaris, S., Van Acker, S., and Zaihrayeu, I.: State of the Art on Ontology

Alignment. (2004)
3. Schwering, A.: Hybrid Model for Semantic Similarity Measurement. OTM Conferences. LNCS Vol. 3761 Springer (2005) 1449-1465
4. Usanavasin, S., Takada, S., and Doi, N.: Semantic Web Services Discovery in Multi-ontology Environment. LNCS Vol. 3762 Springer-Verlag Berlin Heidelberg (2005) 59-68
5. Hau, J., Lee, W., and Darlington, J.: A Semantic Similarity Measure for Semantic Web Services. Web Service Semantics:Towards Dynamic Business Integration (2005)
6. Albertoni, R., Bertone, A., and De Martino, M.: Semantic Analysis of Categorical Metadata to Search for Geographic Information, Proceedings Sixteenth International Workshop on Database and Expert Systems Applications (2005) 453-457
7. Rada, R., Mili, H., Bicknell, E., and Blettner, M.: Development and application of a metric on semantic nets. Systems, Man and Cybernetics, IEEE Transactions on. Vol. 19[1]. (1989) 17-30
8. Li, Y., Bandar, Z., and McLean, D.: An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. IEEE Trans.Knowl.Data Eng. Vol. 15 (2003) 871-882
9. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy (1995) 448-453
10. Lin, D.: An Information-Theoretic Definition of Similarity San Francisco, CA, USA. ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. (1998) 296-304
11. Rodriguez, M. A. and Egenhofer, M. J.: Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. International Journal of Geographical Information Science Vol. 18[3] (2004) 229-256
12. Tversky, A.: Features of similarity. Psycological Review. Vol. 84[4]. (1977) 327-352
13. Gädenfors, P.: How to make the semantic web more semantic. Formal Ontology in Information System. IOS Press (2004) 17-34
14. Maedche, A. and Zacharias, V.: Clustering Ontology Based Metadata in the Semantic Web. (2002)
15. Ehrig, M., Haase, P., Stojanovic, N., and Hefke, M.: Similarity for Ontologies - A Comprehensive Framework. 13th European Conference on Information Systems. (2005)