The original publication is available at www.springerlink.com.
Albertoni R. and De Martino M. ,**Semantic Similarity of Ontology Instances Tailored on the Application Context,** ODBASE- OTM Conferences, LNCS Vol. 4275, pp. 1020-1038 (2006)

Semantic Similarity of Ontology Instances Tailored on the Application Context

Riccardo Albertoni, Monica De Martino

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

Abstract. The paper proposes a framework to assess the semantic similarity among instances within 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 combining and extending existing similarity measures and tailoring them according to the criteria induced by the context. Experiments and evaluation of the similarity assessment are provided.

1 Introduction

In this decade, the ontologies have been imposing in the computer science as artefact to represent explicitly shared conceptualisations. A remarkable research effort has been spent to develop new ontology languages, proper reasoning mechanisms and correlated management tools. Less attention has been posed instead on the similarity among the ontology instances. Methods to assess similarity among instances are needed to exploit the knowledge modelled in the ontology in different research fields pertaining the Knowledge Management such as Data Mining and Information Visualization. They should consider as much as possible the implicit information encoded in the ontology as they provide useful hints to define the similarity. Moreover, they should be sensible to specific contexts inasmuch as different contexts induce different criteria of similarity.

So far, the most of research activity pertaining to similarity and ontologies has been carried out within the field of ontology alignment or to assess the similarity among concepts. Unfortunately, all these methods result inappropriate for the similarity among instances. On the one hand the similarities for the ontology alignment strongly focus on the comparison of the structural parts of distinct ontologies and their application to assess the similarity among instances might result misleading. On the other hand, the concepts' similarities mainly deal with lexicographic database ignoring the comparison of the instances values. Apart from them, few methods to assess similarities among instances have been proposed. Unfortunately these methods rarely take into account the different hints hidden in the ontology and they do not

consider that the ontology entities differently concur in the similarity assessment according to the application contexts.
To overcome the limitations mentioned above the paper proposes a framework to assess the semantic similarity among instances. Its contribution is twofold. Firstly, the framework provides a measurement of semantic similarity more sensitive to the hints hidden in the ontology. It is defined by an amalgamation function, which combines and extends different similarities already defined in literature: it takes into account both the structural comparison between two instances in terms of the classes that the instances belong to, and the instances comparison in term of their attributes and relations. Secondly, the framework provides the parametric evaluation of the similarity with respect to different applications. The application induces the criteria of similarity which are explicitly formalized in the application context. An application context models the importance of the entities, which concur in the assessment of similarity, and the operation used to compare the instances. The parametric evaluation allows to tailor the similarity assessment to specific application contexts, but also to obtain different similarity assessments employing the same ontology.
The paper is organised as follows. In the first section, we illustrate the main principle of the approach. Then a formalization of the similarity criteria induced by the context is proposed. The remaining sections are devoted to the definition of the similarity functions which characterise our method followed by two experiments and an evaluation of the results. At the end, we evaluate the related works underlining our contributions.

2 Semantic Similarity Method

The paper proposes a semantic similarity among instances within an ontology taking into account the different hints hidden in the ontology and the application context. As the hints that can be considered largely depend on the level of formality of the ontology model adopted, it is important to state clearly to which ontology model a similarity method is referring. In the paper, the ontology model with data type defined by Ehrig et al.[1] is considered.
Definition 1: Ontology with Data Type. *An Ontology with data type is a structure* $O := (C, T, \leq_c, R, A, \sigma_R, \sigma_A, \leq_R, \leq_A, I, V, l_C, l_T, l_R, l_A)$ *where C,T,R,A,I,V are disjointed sets respectively of classes, data types, binary relations, attributes, instances and data values, and the relations and functions are defined as follows:*

$\leq_C$	the partial order on C, which defines the classes hierarchy,
$\leq_R$	the partial order on R which defines the relation hierarchy,
$\leq_A$	the partial order on A which defines the attribute hierarchy,
$\sigma_R : R \to C \times C$	the function that provides the signature for each relation,
$\sigma_A : A \to C \times T$	the function that provides the signature for each attribute,
$l_C : C \to 2^I$	the function called class instantiation,
$l_T : T \to 2^V$	the function called data type instantiation,
$l_R : R \to 2^{I \times I}$	the function called relation instantiation,
$l_A : A \to 2^{I \times V}$	the function called attribute instantiation.

Two kinds of similarity exist with symmetric or with asymmetric properties. 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*

An asymmetric normalized similarity is a function $\overline{S} : IxI \rightarrow [0,1]$ that does not satisfy the symmetric axiom. The preference between symmetric and asymmetric similarity mainly depends on the application scenario, there is no a-priori reason to formulate this choice. A complete framework to assess the semantic similarity should provide both of them. In the paper only the asymmetric similarity is described due to lack of space.

The proposed model adopts the schematisation of the similarity framework defined by Ehrig et.al. [1]: they structure 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 et al. is suitable to support the ontology similarity as well as instances similarity.

Our contribution with respect to the framework defined by Ehrig et al. is mainly in the definition of a *context layer* including an accurate formalization of the criteria to tailor the similarity with respect to a context and in the definition of an *ontology layer* explicitly parameterised according to these criteria. Concerning the data and domain knowledge layers the paper adopts a replica of what is illustrated in [1].

The formalization of the criteria of similarity induced by the 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 the following amalgamation function ($\overline{Sim}$) which aggregates two similarity functions defined in the ontology layer named *external similarity* ($\overline{ExternSim}$) and *extensional similarity* ($\overline{ExtensSim}$). The external similarity performs a structural comparison between two instances $i_1 \in l_c(c_1)$, $i_2 \in l_c(c_2)$ in terms of the classes c_1, c_2 the instances belong to, whereas the extensional similarity performs the instances comparison in term of their attributes and relations.

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

$w_{ExternSim}$ and $w_{ExtensSim}$ are the weights to balance the functions importance. By default they are equal to 1\2. In the below sections the Context Layer is described as well as the two similarities $\overline{ExternSim}$ and $\overline{ExtensSim}$.

3 Context Layer

The context layer, according to Ehrig at al. [1], describes how the ontology entities concur in different contexts. The paper adopts this point of view. However it aims to formalize the application context in the sense of modelling the criteria of similarity induced by the context. This design choice does not hamper to define eventually a generic description of context and then to determine automatically which criteria would have been suitable for a given context. Rather, it allows to calculate directly the similarity acting on the criteria especially when it is necessary to refine them. In the following we underlay the importance of this formalization and we provide it.

3.1 Motivation Behind the Application Context Formalization

The application context provides the knowledge to formalise the criteria of similarity induced by the application. Criteria are context-dependent as the context influences both the choice of classes, attributes and relations to be considered in the similarity assessment and the operations to compare them.
We describe the motivation behind the proposed formalization through an example. Let consider a simplified version of the ontology KA[1] that defines concepts from academic research (Fig 1) and focus on the two applications: "comparison of the members of the research staff according to their working experience" and "comparison of the members of the research staff with respect to their research interest". Two distinct application contexts can be induced according the applications:

- "Exp" induced by the comparison of the members of the research staff according to their working experience. The similarity among the members of the research staff (instances of the class *ResearchStaff*[2]) is roughly assessed considering the member's age (the attribute *age* inherited by the class *Person*), the number of projects and publications a researcher has worked on (the number of instances reachable through the relation *publication* and relation *workAtProject* inherited by *Staff*).
- "Int" induced by the comparison of the members of the research staff with respect to their research interest. The researchers can be compared with respect to their interest (instances reachable through the relation *interest*), and again the publications (instances reachable through the relation *publications*), the projects (instances reachable through the relation *workAtProject*).

Analysing these examples the follows considerations can be pointed out:
1. the similarity between two instances can depend on the comparison of their related instances: the researchers are compared with respect to the instances of the class *Publication* connected through the relation *publications*;
2. the attributes and relations of the instances can differently contribute in the evaluation according to the context: the attribute *age* of the researchers is functional in the first application but it might not be interesting in the second; the relations

[1] http://protege.stanford.edu/plugins/owl/owl-library/ka.owl
[2] The italics is used to explicit the reference to the entities (attributes, relations, classes) of the ontology in Fig 1.

publication and *workAtProject* are included in both the application contexts but using different operator of comparison: in the first case just the number of instances is important whereas in the latter the related instances have to be compared;

3. the ontology entities can be considered recursively in the similarity evaluation: in the context "Int" the members' research topic (instances of *ResearchTopic* reachable navigating through the relation *ResearchStaff->interest*[3]*)* are considered and their related topics (instances of *ResearchTopic* reachable via *ResearchStaff->interest->relatedTopic*) are recursively compared to assess the similarity of distinct topics;

4. the classes' attributes and relations can differently contribute in the evaluation according to the recursion level of the assessment: in the second application the attribute *topicName* and the relation *relatedTopic* can be considered at the first level of recursion to assess the similarity between *researchTopic*. By navigating the relation *relatedTopic* it is possible to apply another step of recursion, and here the similarity criteria can be different from the previous ones, for example in order to limit the computational cost and stop the recursion, only the *topicName* or the instances identifier could be adopted to compare the *relatedTopic*.

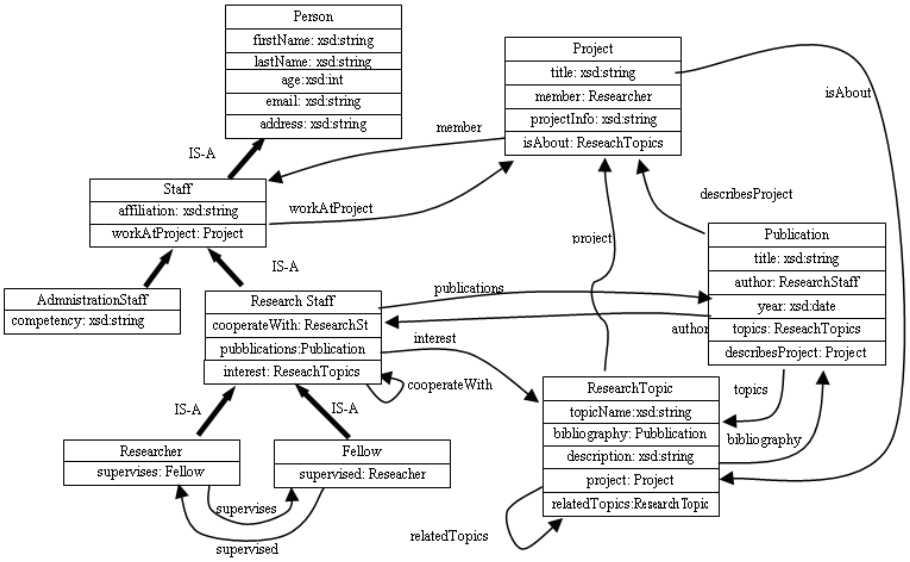

Fig. 1. Ontology defining concepts related to the academic research

As pointed out in the second remark, there are different operations that can be used to compare the ontology entities:

- operation based on the "cardinality" of the attributes or 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 in the first context **"**Exp" two researchers are similar if they have a similar "number" of publications;

[3] The arrow is used to indicate the navigation through a relation, for example *A->B->C* means that starting from the class *A* we navigate through the relations B and C.

- operation based on the "intersection" between sets of attributes or relations: the similarity is assessed according to the number of elements they have in common. For example in the context "Int" the more papers two researchers share, the more their interests are similar;
- operation based on the "similarity" of attributes and relations: the similarity is assessed in terms of similarity of the attributes values and related instances. For example, in the context **"**Int**"** two researchers are similar if they have "similar" research topics.

The example evidences that an accurate formalism is needed to properly express the criteria which might arise from different application contexts. The formalization has to model the attributes and relations as well as the operation to compare their values. Moreover, as noticed in the fourth remark also the level of recursion of the similarity assessment has to be considered.

3.2 Application context formalization

The formalization provided in the sequel represents the restrictions that the application context must adhere to. An ontology engineer is expected to provide the application context according to specific application needs. The formalization relies on the concepts of "sequence of elements belonging to a set X" which formalizes generic sequences of elements and "path of recursion of length i" to track the recursion during the similarity assessment. In particular, a "path of recursion" represents the recursion in terms of sequence of relations used to navigate the ontology.

The application context function (*AC*) is defined inductively on the length of the path of recursion. It returns the set of attributes and relations as well as the operations to be used in the similarity assessment. The considered operations are those illustrated in the previous paragraph and named respectively *Count* to evaluate the cardinality, *Inter* to evaluate the intersection, *Simil* to evaluate the similarity.

Definition 2: Sequences of a Set X. *Given a set X, a sequence s of elements of X with length n is defined by the function* $s:[1,..,n]\to X, n\in N^+$ *and represented in simple way by the list [s(1),..,s(n)].*

Let mark $S_X^n=\{s\mid s:[1,n]\to X\}$ 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.

Let define in Table 1 the polymorphic functions which identify specific sets of entities in the ontology model.

Table 1. List of functions defining specific sets of elements in the ontology model.

$\delta_a: C\to 2^A; \delta_a(c)=\{a:A\mid \exists\, t\in T, \sigma_A(a)=(c,t)\}$	set of attributes of $c\in$ C ,
$\delta_a: R\to 2^A; \delta_a(r)=\{a:A\mid \exists c,c'\in C\ \exists\, t\in T\ \sigma_R(r)=(c,c') \wedge\sigma_A(a)=(c',t)\}$	set of attributes of the classes which are reachable through the relation r∈ R,
$\delta_r: C\to 2^R; \delta_r(c)=\{r:R\mid \exists\, c'\in C, \sigma_R(r)=(c,c')\}$	set of relations of $c\in$C ,
$\delta_c: R\to 2^C; \delta_c(r)=\{c':C\mid \exists\, c\in C\ \ \sigma_R(r)=(c,c')\}$	set of concepts reachable through r∈ R,
$\delta_r: R\to 2^R; \delta_r(r)=\{r':R\mid \exists\, c\in C, \exists\, c'\in\delta_C(r); \sigma_R(r')=(c',c)\}$	set of relations of the concepts reachable through r ,
$\delta_c: C\to 2^C; \delta_c(c)=\{c':C\mid \exists\, r\in\delta_r(c); \sigma_R(r)=(c,c')\}$	set of concepts related to $c\in$C through a relation.

Definition 3: Path of Recursion. *A path of recursion p with length i is a sequence whose first element is a class and the other are relations recursively reachable from the class:* $p \in S^i_{C \cup R} \mid p(1) \in C \wedge \forall j \in [2,i]\ p(j) \in R \wedge p(j) \in \delta_r(p(j-1))$.

For example of path of recursion with length longer than three is a path which starts from a class (p(1)) and continues in one of its relations as second element p(2), in one of the relations of the class reachable from p(2) as third element p(3) and so on. In general, a path of recursion p represents a path to be followed to assess the similarity recursively. The recursion expressed in the previous paragraph in the context "Int" as *ResearchStaff->interest->relatedTopic* is formalised with the path of recursion [ResearchStaff, interest, relatedTopic].

Let name P^i the set of all paths of recursion with length i and P the set of all paths of recursion $P= \cup_{i \in N} P^i$.

Definition 4: Application Context AC. *Given the set P of paths of recursion,* $L = \{Count, Inter, Simil\}$ *the set of operations adopted, an application context is defined by a partial function AC having signature* $AC : P \rightarrow (2^{A \times L}) \times (2^{R \times L})$ *returning the attributes and relations as well as the operations to perform their comparison.*

In particular, each application context AC is characterised by two operators $AC_A : P \rightarrow 2^{A \times L}$ and $AC_R : P \rightarrow 2^{R \times L}$ which return respectively the part of context AC related to the attributes and the relations. Formally $\forall p \in P\, AC(p) = (AC_A(p),\ AC_R(p))$ and $AC_A(p)$ and $AC_R(p)$ are set of pairs $\{(e_1,o_1), (e_2,o_2),\ldots, (e_i,o_i),\ldots,(e_n,o_n)\}$ n $\in$N where e_i is respectively the attribute or the relation relevant to define the similarity criteria and $o_i \in$L is the operation to be used in the comparison.

We provide two examples of AC formalization referring to the two application contexts "Exp", "Int" mentioned in the previous paragraph.

Example 1. Let formalise the application context "Exp" with AC_{Exp} to assess the similarity among the members of a research staff according to their experience. We consider the set of paths of recursion {[ReasearchStaff], [Reasearch], [Fellow]} and we compare them according to the age similarity, the number of publications and projects. Thus AC_{Exp} is defined by:

$$[\text{ResearchStaff}] \xrightarrow{AC_{Exp}} \{\{(\text{age},\text{Simil})\},\{(\text{publications},\text{Count}),(\text{workAtProject},\text{Count})\}\} \quad \textbf{(2)}$$

$$[\text{Researcher}] \xrightarrow{AC_{Exp}} \{\{(\text{age},\text{Simil})\},\{(\text{publications},\text{Count}),(\text{workAtProject},\text{Count})\}\}$$

$$[\text{Fellow}] \xrightarrow{AC_{Exp}} \{\{(\text{age},\text{Simil})\},\{(\text{publications},\text{Count}),(\text{workAtProject},\text{Count})\}\}$$

An example of AC_R is {(publication,Count),(workAtProject,Count)} while an example of AC_A is {(age,Simil)}.

Let note that [Researcher] and [Fellow] belong to the set of path of recursion considered in AC_{Exp} because their instances are also instance of *ResearchStaff*. The application context can be expressed in a more compact way assuming that whenever a context is not defined for a class but is defined for its super class, the comparison criteria defined for a super class are by default inherited by the subclasses. According to this assumption AC_{Exp} can be expressed through,

$$[\text{ResearchStaff}] \xrightarrow{AC_{Exp}} \{\{\text{age},\text{Simil}\},\{(\text{publications},\text{Count}),(\text{workAtProject},\text{Count})\}\} \quad \textbf{(3)}$$

Example 2. Let formalise the application context "Int" to assess the similarity among the members of a research staff according to their research interest. The similarity is computed considering the set of path of recursion {[ResearchStaff],[ResearchStaff,

interest]}. The researchers are compared considering common publications, common projects or similar interests. A compact formalization for "Int" is defined by AC_{Int}:

$$[\text{ResearchStaff}] \xrightarrow{AC_{Int}} \{\{\phi\}, \{(\text{publications}, \text{Inter}), (\text{workAtProject}, \text{Inter}), (\text{interest}, \text{Simil})\}\} \quad \textbf{(4)}$$

$$[\text{ResearchStaff}, \text{interest}] \xrightarrow{AC_{Int}} \{\{\text{topicName}, \text{Inter}\}, \{(\text{relatedTopics}, \text{Inter})\}\}$$

Let note that the researchers are compared recursively: [ReseachStaff,interest] is the path of recursion to navigating the ontology from *ResearchStaff* to *ResearchTopic* via the relation *interest.* The interests are compared with respect to both their *topicName* and their *relatedTopic*, thus two *ResearchTopic*(s) having distinct *topicName* but some *relatedTopic* in common are not considered completely dissimilar.

The image of an AC function can be further characterized:

1. For a path of recursion p, AC has to return only the attributes and relations belonging to the classes reached through p. For example, considering the ontology in fig 1 and the path of recursion [ReseachStaff,interest] it is expected that only the attributes and relations belonging to the class *ResearchTopic* reachable via [ReseachStaff,interest], can be identified by AC([ReseachStaff,interest]). Attributes or relations (as *age, publications, etc)* which do not belong to *ResearchTopic* define an incorrect application context.
2. Given a path of recursion p, an attribute or a relation can appear in the context image at most one time. In other words, given a path of recursion it is not possible to associate two distinct operations to the same relation or attribute. For example the following application context definition is not correct as *interest* is specified twice

$$[\text{ResearchStaff}] \longrightarrow \{\{\phi\}, \{(\text{publications}, \text{Inter}), (\text{interest}, \text{Simil}), (\text{interest}, \text{Inter})\} \quad \textbf{(5)}$$

4 Ontology Layer

The ontology layer defines the asymmetric similarity functions $\overline{ExternSim}$ and $\overline{ExtensSim}$ which compose the amalgamation function (formula 1). The "external similarity" $\overline{ExternSim}$ measures the similarity at the level of the ontology schema computing a structural comparison of the instances: 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 similarity" $\overline{ExtensSim}$ compares the extension of the ontology entities: the similarity is assessed by computing the comparison of the attributes and relations of the instances.

At the ontology layer additional hypotheses are assumed:

- 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 do not have any common super-class different from *Thing,* their similarity is equal to 0.
- The least upper bound (*lub)* between c_1 and c_2, is unique and it is c_2 if c_1 IS-A c_2, or c_1 if c_2 IS-A c_1, otherwise the immediate super-class of c_1 and c_2 that subsumes both classes.

The aim is to force the *lub* to be a sort of "template class" which can be adopted to perform the comparison of the instances whenever the instances belong to distinct

classes. Referring to the ontology in Fig 1, it can be appropriate to compare two instances belonging respectively to *AdministratorStaff* and *ResearchStaff* as they are both a kind of staff and *Staff* is their *lub*. However, it does not make sense to evaluate the similarity between two instances belonging to *Publication* and to *Staff,* because they are intimately different: actually there is not any *lub* available for them.
Whenever a *lub* x between the two classes exists, the path of recursion [x] is the starting path in the recursive evaluation of the similarity.

4.1 External Similarity

The external similarity ($\overline{ExternSim}$) performs the structural comparison between two instances i_1, i_2 in terms of the classes c_1, c_2 the instances belong to: more formally $\overline{ExternSim}(i_1,i_2) = \overline{ExternSim}(c_1,c_2)$ where $i_1 \in l_c(c_1), i_2 \in l_c(c_2)$.
In the paper the external similarity function is defined starting from the similarities proposed by Maedche and Zacharias [2] and Rodriguez and Egenhofer [3]. The structural comparison is performed by two similarity evaluations:

- **Class Matching** which is based on the distance between the classes c_1, c_2 and their depth respect to the hierarchy induced by $\leq_C$.
- **Slot Matching** which is based on the number of attributes and relations shared by the classes c_1, c_2 and the overall number of their attributes and relations. Then two classes having a plenty of attributes/relations, some of whose are in common, are less similar than two classes having less attributes but the same number of common attributes/relations.

Both similarities are needed to evaluate the similarity with respect to the ontology structure with success. For example, let us consider the ontology schema in Fig. 2 and let compare an instance of the class D with an instance of the class E.

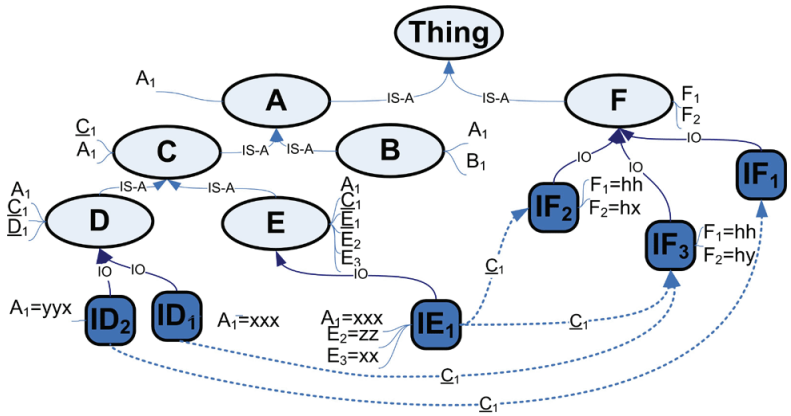

Fig. 2. Class hierarchy example: A, B, C, D, E, F are classes, A_1,B_1, E_2, E_3, F_1, F_2 are attributes, $\underline{C}_1$, $\underline{D}_1$, $\underline{E}_1$ are relations, ID_1, ID_2, IE_1, IF_1, IF_2, IF_3 are instances

They are quite similar with respect to the class matching but less similar with respect to the slot matching. At the fact, the sets of IS-A relations joining the classes D and E to *Thing* are largely shared. However, from the point of view of the slots, D and E share only the attribute A_1 and relation $\underline{C}_1$ and they differ with respect to the others. Likewise it would be easy to show an example of two classes similar with respect to the slots matching and dissimilar according to the class matching.

Definition 5: ExternSim similarity. *The similarity between two classes according to the external comparison is defined by:*

$$\overline{ExternSim}(c_1,c_2)=\begin{cases}1 & if\ \ c_1=c_2\\ \dfrac{w_{SM}*\overline{SM}(c_1,c_2)+w_{CM}*\overline{CM}(c_1,c_2)}{w_{SM}+w_{CM}} & Otherwise\end{cases} \qquad \textbf{(6)}$$

where ($\overline{SM}$) is the Slots Matching, ($\overline{CM}$) is the Classes Matching and w_{SM}, w_{CM} the respectively weights in the range [0,1].
w_{SM} and w_{CM} are defined for the purpose of this paper equal to 1\2.

4.1.1 Class Matching

Classes Matching is evaluated in terms of distance of the classes with respect to the IS-A hierarchy. The distance is based on the concept of Upwards Cotopy (*UC*)[2]. We define an asymmetric similarity adapting the symmetric definition of CM in [2] .

Definition 6: Upward Cotopy (UC). *The Upward Cotopy of a set of classes C with the associated partial order $\leq_C$ is:*

$$UC_{\leq_C}(c_i) := \{c_j \in C \mid (c_i \leq_C c_j) \vee c_i = c_j\} \qquad \textbf{(7)}$$

It is the set of classes composing the path to reach from c_i the furthest super-class (*Thing*) of the IS-A hierarchy: for example considering the class D in Fig. 2 $UC_{\leq_C}(D)=\{\mathrm{D},\mathrm{C},\mathrm{A},\mathrm{Thing}\}$.

Definition 7: Asymmetric Class Matching. *Given two classes c_1, c_2, the Upward Cotopy $UC_{\leq_C}(c_i)$, the asymmetric Class Matching is defined by:*

$$\overline{CM}(c_1,c_2) := \frac{\left|UC_{\leq_C}(c_1)\cap UC_{\leq_C}(c_2)\right|}{\left|UC_{\leq_C}(c_1)\right|} \qquad \textbf{(8)}$$

$\overline{CM}$ between two classes depends on the number of their common classes in the hierarchy. Let note that the class matching is asymmetric, for example referring to Fig. 2, $\overline{CM}(B,D)=2/3$ but $\overline{CM}(D,B)=2/4$. Moreover, it is important to note that $\overline{CM}(A,D)=1$, the rationale behind this choice of design is that the instances of D are suitable as instances of A.

4.1.2 Slot Matching

Slot Matching is defined by the slots (attributes and relations) shared by the two classes. We refer to the similarity proposed by Rodriguez and Egenhofer [3] based on the concept of distinguishing features employed to differentiate subclasses from their super-class. In their proposal, different kinds of distinguishing features are considered (i.e. functionalities, and parts) but no one coincides immediately with the native

entities in our ontology model. Of course it would be possible to manually annotate the classes adding the distinguishing features but we prefer to focus on what is already available in the adopted ontology model. Therefore only attributes and relations are mapped as two kinds of distinguishing features.

Definition 8: Slot Matching. *Given two classes c_1,c_2, two kinds of distinguishing 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.*

$$\overline{SM}(c_1,c_2) = \omega_a \cdot \overline{S}_a(c_1,c_2) + \omega_r \cdot \overline{S}_r(c_1,c_2) \quad \textbf{(9)}$$

The sum of weights is expected to be equal to 1, and by default we assume to be w_a=w_r=1/2. The two slot matching $\overline{S}_a$ and $\overline{S}_r$ rely on the definitions of *slot importance* as defined in the following.

Definition 9: Function of "Slot Importance" α. *Let c_1, c_2, be two distinct classes, d the class distance $d(c_1,c_2)$ in term of the number of edges in a IS-A hierarchy, α is the function that evaluates the importance of the difference between the two classes.*

$$\alpha(c_1,c_2) = \begin{cases} \dfrac{d(c_1,\text{lub}(c_1,c_2))}{d(c_1,c_2)} & d(c_1,\text{lub}(c_1,c_2)) \le d(c_2,\text{lub}(c_1,c_2)) \\ 1 - \dfrac{d(c_1,\text{lub}(c_1,c_2))}{d(c_1,c_2)} & d(c_1,\text{lub}(c_1,c_2)) > d(c_2,\text{lub}(c_1,c_2)) \end{cases} \quad \textbf{(10)}$$

Where $d(c_1,c_2) = d(c_1,\text{lub}(c_1,c_2)) + d(c_2,\text{lub}(c_1,c_2))$.

$\alpha(c_1, c_2)$ is a value in the ranges [0,0.5]. Referring to the image Fig. 2, α(D,C) is equal to zero because the lub between D and C is C itself, d(C,D)=1 and d(C,C)=0. Whereas α(D,E) is equal to 0.5 because the lub is still C, and d(D,E)=2.

Definition 10: Slot Matching according to the kind of distinguishing feature t. *Given two classes c_1 (target) and c_2, (base), t a kind of distinguishing feature (t=a for attributes or t=r for relations), let be C_1^t and C_2^t the sets of distinguishing features of type t respectively of c_1 and c_2; the Slot Matching $\overline{S}_t(c_1,c_2)$ is defined by:*

$$\overline{S}_t(c_1,c_2) = \frac{\left|C_1^t \cap C_2^t\right|}{\left|C_1^t \cap C_2^t\right| + \alpha(c_1,c_2)\left|C_1^t \setminus C_2^t\right| + (1-\alpha(c_1,c_2))\left|C_2^t \setminus C_1^t\right|} \quad \textbf{(11)}$$

According to the ontology in Fig. 2, considering the classes D and E their sets of distinguishing features of type relation are D^r ={$\underline{C}_1$, $\underline{D}_1$} and E^r={$\underline{C}_1$, $\underline{E}_1$} and α(D,E)=0.5; then $\overline{S}_r(D,E) = 0.5$.

In general, whenever α=0.5 the difference of the features of both classes are equally important for the matching: for example it happens when the classes are sisters as in the case of D and E. In the case α=0 only the features that are in c_2 and not in c_1 are important for the matching. In particular it happens, whenever c_2 is the subclass of c_1, in this case the matching is inversely proportional to the higher number of features of c_2 compared to those of c_1 .

4.2 Extensional Similarity

The extension of entities plays a fundamental role in the assessment of the similarity among the instances, it is needed to perform a comparison of the attribute and relation values. For example, in the ontology in Fig. 2 relying only on the structural comparison it is not possible to assess that ID_1 is more similar to IE_1 than to ID_2. The main principle of the proposed extensional similarity between two instances is to consider the lub x of their classes as the common base to compare them when the instances belong to different classes: it is adopted to define the path of recursion [x] from which starts the recursive assessment induced by an application context.
For example, considering the instances ID_1 and IE_1 in Fig. 2, the class C is their lub. Then the initial path of recursion from which to start the similarity assessment is [C]. Let us suppose to have already defined an application context as the follow [C]→{{(A_1,*Iter*)},{($\underline{C_1}$, *Simil*)}}; [C,$\underline{C_1}$]→{{(F_1,*Simil*)},{}}. The computation starts from the values of attribute A_1 for the instances ID_1 and IE_1, then through the relation $\underline{C_1}$ the new path of recursion [C,$\underline{C_1}$] is considered to compare the instances related to $\underline{IE_1}$ and $\underline{ID_1}$ with respect to the values of the attribute F_1.
The extensional comparison is characterised by two similarities functions: a function based on the comparison of the attributes of the instances and a function based on the comparison of the relations of the instances.

Definition 11: Extensional Asymmetric Similarity. *Given two instances $i_1 \in l_c(c_1)$, $i_2 \in l_c(c_2)$, $c=lub(c_1,c_2)$, $p=[c]$ 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 their attributes and their relations. The extensional similarity with asymmetric property is defined:*

$$\overline{ExtensSim}(i_1,i_2)=\begin{cases}1 & i_1=i_2\\ \overline{Sim}_I^p(i_1,i_2) & Otherwise\end{cases} \qquad \textbf{(12)}$$

Where $\overline{Sim}_I^p(i_1,i_2)$ is defined by:

$$\overline{Sim}_I^p(i_1,i_2)=\frac{\sum_{a\in\delta_a(c)}\overline{Sim}_a^p(i_1,i_2)+\sum_{r\in\delta_r(c)}\overline{Sim}_r^p(i_1,i_2)}{|\delta_a(c)|+|\delta_r(c)|} \qquad \textbf{(13)}$$

Let note that the index p is a kind of stack of recursion adopted to track the navigation of relations whenever the similarity among instances is recursively defined in terms of the 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 as following.

Definition 12: Similarity on Attributes and Relations. *Given two instances $i_1 \in l_c(c_1)$, $i_2 \in l_c(c_2)$, $c=lub(c_1,c_2)$, $p=[c]$ a path of recursion, X a placeholder for the "A" or "R", $x \in A \cup R$ let be:*

- $i_A(i)=\{v\in V \mid (i,v)\in l_A(a), \exists y\in C\ s.t.\ \sigma_A(a)=(y,T)\wedge l_T(T)=2^V\}$ *the set of values assumed by the instance i for the attribute a,*
- $i_R(i)=\{i'\in l_c(c') \mid \exists c\ i\in l_c(c)\exists c'\ s.t.\ \sigma_R(r)\in(c,c')\wedge(i,i')\in l_R(r)\}$ *the set of instances related to the instance i by the relation r,*

- *AC the application context defined* according *to the restrictions defined in paragraph 3.2*
- $F_X = \{g : i_X(i_1) \rightarrow i_X(i_2) \mid g \text{ is partial and bijective}\}$

The similarity between instances according to their attributes or relations is defined:

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

The above formulas are designed to be asymmetric. Asymmetry is used to ensure that considering the relations and attributes selected by the application context, if an instance i_1 has at least the same attribute and relation values of i_2 then the extensional similarity between i_2 and i_1 is equal to one.

The method compute $\overline{\mathrm{Sim}}_x^p$ selecting one of the above formulas according to the definition of *AC*: if AC returns a relation or attribute having as operation *Count* the third formula is adopted, as operation *Inter* the fourth is considered, and so on. The fifth formula is adopted whenever the AC returns an attribute whose operation is *Simil*. In this case, the comparison of attribute values rely on $\overline{Sim}_T^a$ which defines the similarity for values of the attribute a having data type T. $\overline{Sim}_T^a$ is provided by the data layer as suggested by [1]. The set of partial functions in F are employed to represent the possible matching among set of values when instances have relations or attributes with multiple values. For example, the instance IE_1 has IF_3 and IF_2 related via $\underline{C}_1$, ID_1 has IF_3, when IE_1 and ID_1 are compared two possible partial and bijective functions f_1 and f_2 can be considered between the instances related to IE_1 and ID_1: f_1:$IF_2 \rightarrow IF_3$ and f_2:$IF_3 \rightarrow IF_3$.

It is important to note that each time the similarity is assessed in terms of related instances (whenever (r,*Simil*) $\in$ $AC_R(p)$) the relation r that is followed to reach the related instances is added to the path of recursion. Thus during the recursive assessment the AC is always worked out on the most updated path of recursion.

5 Experiment and Evaluation

The similarity assessment among the research staff working at the Institute (CNR-IMATI-GE) is considered as application case to evaluate the proposed method. Two

experiments are performed considering the contexts “Exp”, “Int**”** mentioned in paragraph 4.1. Eighteen members of the research staff are considered; the information related to their projects, journal publications and research interests are inserted as instances in the ontology depicted in Fig. 1 according to what is published at the IMATI web site[4]. The ontology is expressed in OWL paying attention to adopt only the language constructs that allow to remain within the ontology model considered in definition 1. The resulting ontology is available at the web site [4]. Our method is implemented in JAVA and is tested on this ontology.

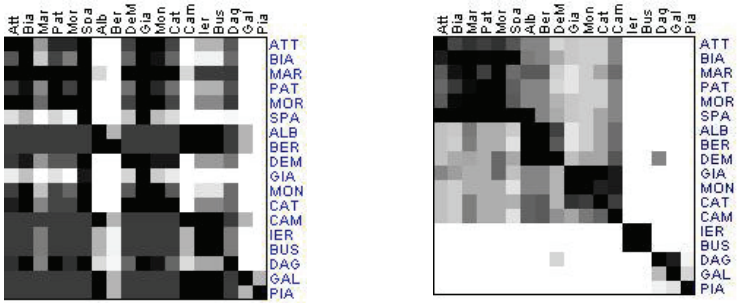

Fig. 3. (a) Similarity matrix for context “Exp”; (b) Similarity matrix for context “Int”

Using the formalization of the two application contexts AC_{Int} e AC_{Exp} previously defined (formulas (3), (4)) we have computed the similarity through the proposed framework. The results are represented by the similarity matrices in Fig. 3: (a) is the result related the context “Exp**”** and (b) is the result related to the context “Int”. Each column j and each row i of the matrix represent a member of the research staff (identified by the first three letter of his name). The grey level of the pixel (i,j) represents the similarity value (Sim(i,j)) between the two members located at the row i and columns j: the darker is the colour the more similar are the two researchers.
Analysing the similarity matrices it is easy to realize that they are asymmetric; this confirms that the proposed model assesses an asymmetric similarity. Comparing the two matrices, it stands out how they are different: it is evident that the two contexts induce completely different similarity values. For example, “Dag” results very similar to “Bia” with respect to their experience (black pixel in Fig. 3.a), but they are no similar with respect to their research interest (white pixel in Fig. 3.b). Moreover sim(Dag,Bia)>sim(Bia, Dag) in Fig. 3.a means that Bia has at least the experience of Dag and she/he can be considered similar to Dag (if somebody with the Dag experience is searched), but the inverse is not true.
Two kind of evaluations of the result concerning the similarity obtained with respect to the research interest (Fig. 3.b) are performed.
The first evaluation is based on the concept of recall and precision calculated considering the same adaptation of recall and precision made by [5]. More precisely, considering an entity x the recall and precision are defined respectively as $(A \cap B)/A$, $(A \cap B)/B$ where A is the set of entities expected to be similar to x, and B is the set of

[4] http://www.ge.imati.cnr.it

similar entity calculated by a model. A critical issue in the similarity evaluation is to have a ground truth with respect to comparing the results obtained. We face this problem referring to the research staff of our institute and considering "similar" two members of the same research group. At the fact at IMATI researchers and fellows are grouped in three main research groups and one of those is composed by further three sub-groups. Then we consider the research staff as split in five groups. For each member i, A is the set of members of his research group while B is composed by the first n members retrieved by the model. For each group we have calculated recall and precision considering as "n" the smaller number of member needed to obtain a recall of 100%, and then we have evaluated the precision. The average recall is estimated equal to 100% with a precision of 95%. These results are quite encouraging: the recall equal to 100% demonstrates that for each research group the similarity is able to rank all the expected members while the precision equal to 95% means that the average number of outsiders to be considered to rank all group members is equal to 5%.

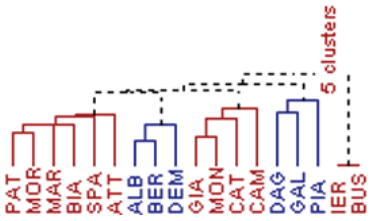

Fig. 4. The dendrogram obtained through the hierarchical gene clustering

We have performed a second evaluation according to the context "Int" using a data mining application. For each researcher and fellow we have computed his similarity with respect to the other members applying our method. In this way, we associate to each research staff member a string of values which correspond to his relative distance from the other members. The strings correspond to the rows of the similarity matrix (**Fig. 3**.b). Then we have applied a tool to perform the hierarchical clustering among genetic microarray [6] to the set of strings, considering each string as a kind of researcher genetic code. The dendrogram obtained is shown in **Fig. 4**, it recognizes the five clusters which resemble the research group structure of our institute.

6 Related Work

Semantic similarity is intended differently according to the application domain where it is adopted. Currently it is relevant in the ontology alignment [7,8], conceptual retrieval [9] as well as semantic web service discovery and matching [10,11] and it is expected to increase its relevance in framework as for the metadata analysis [12].
We discuss related works grouping them in different tracks according to their purpose and the ontology model they adopt.
Similarities in the Ontology alignment. There are plenty of methods to align ontology, as pointed out by Euzenat et al. [8]. The semantic similarity is adopted in this context to figure out relations among the entities in the ontology schemas. It is employed to compare the name of classes, attributes and relations, determining reasonable

mapping between two distinct ontologies. On the contrary, the method proposed in this paper is specifically designed to assess similarity among instances belonging to the same ontology. Some similarities adopted for the ontology alignment consider quite expressive ontology language, (e.g., [7] focus on a subset of OWL Lite) but they mainly focus on the comparison of the structural aspects of ontology. Due to the different purpose of these methods, they result to be unsuitable to properly solve the similarity among instances.

Concepts similarity in 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 [13] or information theory-based methods [14]. 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 [13]. The information theory-based method defines the similarity of two concepts in terms of the maximum information content of the concept which subsumes them [15,16]. Recently new hybrid approaches have been proposed: Rodriguez and Egenhofer [3] takes advantage from the above methods and adds the idea of features matching introduced by Tversky [17]. Schwering [9] 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 [17] or as geometric space [18]. With respect to the method presented in this paper Rada et al. [13], Resnik [15], Lin [16] work on lexicographic databases where the instances are not considered. If they are adopted as they were originally defined to evaluate the instances similarity they are doomed to fail since they ignore important information provided by instances, attributes and relations. Moreover, Rodriguez and Egenhofer [3] and Schwering [9] use the features or even conceptual spaces, information that are not native in the ontology design and should be manually added. Instead the method proposed here aims at addressing as much as possible the similarity taking advantage from the information that have been already spread in the ontology. Additional information are considered only to perform a tuning of the similarity with respect to different application context.

Similarities which rely on ontology models having instances. Other works define similarity relying on ontology models closer to those adopted in the semantic web standards. On the one hand, Hau et al. [11] 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. On the other hand, Maedche and Zacharias [2] 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. Even if these methods consider ontology models which are more evolved than the taxonomy or terminological ontology, their design ignores the need to tailor the semantic similarity according to specific application contexts. Thus to assess the similarity experimented in this paper, two distinct ontologies need to be defined instead of simply defining two contexts as we do.

Contextual dependent similarity. Some papers combine the context and the similarity. Kashyap and Sheth [19] use the concept of semantic proximity and context to achieve

the interoperability among different databases. The context represents the information useful to determine the semantic relationships between entities belonging to different databases. However they do not define a semantic similarity in the sense we are addressing and the similarity is classified in some discrete value (Semantic Equivalence, Semantic Relevance, Semantic Resemblance, etc). Rodriguez and Egenhofer [3] integrate the contextual information into the similarity model. They define as application domain the set of classes that are subject to the user's interest. As in our proposal, they aim to make the similarity assessment parametric with respect to the considered context. Moreover, differently from our methods they formalise the context rather then the similarity criteria induced by the context.
The discussion of the related works shows that beside semantic similarity is defined from different parties, these definition are far from provide a complete framework as intended in our work: they often have different purposes, they consider simpler ontology model, or they completely ignore the need of tailoring the similarity assessment with respect to specific application context. Of course, some of the mentioned works have been particularly precious in the definition of our proposal. As already mentioned during the presentation of the paper both Maedche and Zacharias [2] and Rodriguez and Egenhofer [3] have strongly inspired the part related to the structural similarity. However, to successfully support our purposes the class slots have been considered as distinguishing features. Furthermore, the methods proposed by Maedche and Zacharias [2] for the class matching defines a similarity which is symmetric, thus we have adapted the original in order to make it asymmetric.

7 Conclusions and Future Work

The paper proposes a framework to assess the semantic similarity among instances within an ontology. It combines and extends different existing similarity methods taking into account as much as possible the hints encoded in the ontology and considering the application context. A formalization of the criteria induced by the application is provided as a mean to parameterise the similarity assessment and to formulate a measurement more sensible to the specific application needs.
The framework is expected to bring a great benefit in the analysis of the ontology driven metadata repository. It provides a flexible solution to tailor the similarity assessments according with the different applications: the same ontology can be employed in different similarity assessments simply defining distinct criteria, and there is no more need of building a different ontology for each similarity assessment.
Nevertheless some research and development issues are still open. For example in the proposed approach the formalization of application context affects only the similarity defined by the extensional comparison. It could be interesting to deepen if the context results also in the external comparison similarity. Moreover, it would be worth to extend the similarity to ontology model towards OWL and to test it in more complex use cases.

Acknowledgements

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

References

1. Ehrig, M., Haase, P., Stojanovic, N., and Hefke, M.: Similarity for Ontologies - A Comprehensive Framework. ECIS 2005. Regensburg, Germany (2005)
2. Maedche, A. and Zacharias, V.: Clustering Ontology Based Metadata in the Semantic Web. PKDD 2002 LNAI Springer-Verlag (2002) 348-360
3. Rodriguez, M. A. and Egenhofer, M. J.: Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. IJGIS. Vol. 18[3]. (2004) 229-256
4. Test Ontology, http://www.ge.imati.cnr.it/ima/personal/albertoni/odbase06p.owl.
5. Rodriguez, M. A. and Egenhofer, M. J.: Determining semantic similarity among entity classes from different ontologies. IEEE Trans.Knowl.Data Eng. Vol. 15[2]. (2003) 442-456
6. Hierarchical Clustering Explorer, 3.0, http://www.cs.umd.edu/hcil/multi-cluster/.
7. Euzenat, J. and Valtchev, P.: Similarity-Based Ontology Alignment in OWL-Lite. ECAI. Valencia, Spain IOS Press (2004) 333-337
8. Euzenat, J., Le Bach, T., and et al.: State of the Art on Ontology Alignment. (2004) http://www.starlab.vub.ac.be/research/projects/knowledgeweb/kweb-223.pdf
9. Schwering, A.: Hybrid Model for Semantic Similarity Measurement. OTM Conferences. LNCS Vol. 3761 Springer-Verlag (2005) 1449-1465
10. Usanavasin, S., Takada, S., and Doi, N.: Semantic Web Services Discovery in Multi-ontology Environment. OTM Workshop 2005 LNCS Vol. 3762 Springer-Verlag (2005) 59-68
11. Hau, J., Lee, W., and Darlington, J.: A Semantic Similarity Measure for Semantic Web Services. Web Service Semantics:Towards Dynamic Business Integration, workshop at WWW 05. (2005)
12. Albertoni, R., Bertone, A., and De Martino, M.: Semantic Analysis of Categorical Metadata to Search for Geographic Information. Proceedings 16th International Workshop on Database and Expert Systems Applications, 2005. IEEE (2005) 453-457
13. Rada, R., Mili, H., Bicknell, E., and Blettner, M.: Development and application of a metric on semantic nets. IEEE Trans.Syst.Man Cybern. Vol. 19[1]. (1989) 17-30
14. Li, Yuhua, Bandar, Zuhair, and McLean, David: An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. IEEE Trans.Knowl.Data Eng. Vol. 15(2003) 871-882
15. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. Proc. of the Fourteenth Int. Joint Conference on Artificial Intelligence (1995) 448-453
16. Lin, D.: An Information-Theoretic Definition of Similarity, Proc. of the Fifteenth Int. Conference on Machine Learning. Morgan Kaufmann (1998) 296-304
17. Tversky, Amos: Features of similarity. Psychological Review. Vol. 84[4]. (1977) 327-352
18. Gädenfors, P.: How to make the semantic web more semantic.FOIS.IOS Press (2004) 17-34
19. Kashyap, Vipul and Sheth, Amit: Semantic and schematic similarities between database objects: a context-based approach. VLDB J. Vol. 5[4]. (1996) 276-304