The role of lexical resources in matching classification schemas

P. Bouquet*, L. Serafini†, S. Zanobini*

*Department of Information and Communication Technology
University of Trento
Via Sommarive, 14 – 38050 Povo (Trento), Italy
{bouquet, zanobini}@dit.unitn.it
† Istituto per la Ricerca Scientifica e Tecnologica
Via Sommarive, 18 – 38050 Povo (Trento), Italy
serafini@itc.it

Abstract

In this paper, we describe the role and the use of WORDNET as an external lexical resource in a methodology for matching hierarchical classifi cation schemas. The main difference between our methodology and others which were presented is that we pay a lot of effort in eliciting the meaning of the structures we match, and we do this by using extensively lexical knowledge about the words occurring in labels. The result of this elicitation process is encoded in a formal language, called WDL (WORDNET Description Logic), which is our proposal for injecting lexical semantics into more standard knowledge representation languages.

1. Introduction

In this paper, we describe the role and the use of external lexical resources in a methodology we developed for discovering semantic mappings between pairs of nodes belonging to heterogeneous hierarchical classification schemas. In the current implementation of our methodology,called CTXMATCH22, the lexical resource we use is WORDNET (Fellbaum,); however, nothing in what we did presupposes an exclusive commitment to WORDNET, which means that – with a minor effort in software development – WORDNET can be integrated or replaced with other lexical resources, if available.

In the area of the Semantic Web, a lot of effort has been devoted to develop methodologies for discovering mappings across heterogeneous conceptual schemas. Schemas can be simple thesauri or taxonomies, or much more complex specifications, like fully fledged formal ontologies[1]. Different techniques are based on very different approaches, ranging from the use of schema matching algorithms to semantic analysis. CTXMATCH2 belongs to the latter category, as maps pairs of nodes from different classification schemas through logical reasoning applied to a formalization of the (intuitive) meaning of nodes. As such, it is based on two main steps: the first, called *meaning elicitation*, takes a classification schema as input and returns a formal representation (currently, a formula of Lexical Description Logic (LDL), which will be described below) of the meaning of each node; the second, called *meaning comparison*, takes as input a pair of nodes of two different schemas (plus additional background knowledge, as we will say) and returns a relation such as $\equiv$ (the two nodes have the same meaning), $\perp$ (the meaning of the two nodes is disjoint), $\sqsubseteq$ (the meaning of the first node is less general than the meaning of the second); this relation is deduced via a standard reasoner for the logic we selected.

In this paper, we illustrate and discuss the crucial role played by lexical resources in our approach to meaning elicitation. Indeed, most existing methods (perhaps, all but CTXMATCH2) tend to focus on the structural properties of the schemas to be matched, and do not attempt any analysis of the labels' meaning. Even the methods which use some lexical resource tend to see it as a mere source of synonyms (which are then used to improve the degree of similarity between structures), and not as a source of valuable knowledge which can be used to compute relations. Instead, CTXMATCH2's most innovative part is a technique for making explicit the meaning of each node label, and then build the meaning of that node in the context of the schema in which it occurs. The paper goes as follows: first we discuss why lexical knowledge is crucial in schema matching; the we show how we used WORDNET to create a formal representation of meaning in a logical language called LDL (Lexical Description Logic), and finally we show how meaning elicitation is used in matching classification schemas.

2. Structural and lexical analysis

The starting point of our work is the observation that for humans the most valuable source of information about the meaning of a node in a classification schema (e.g. the one depicted in Figure 1) is labels. Of course, the structure (e.g. the path in which a node occurs) is also very important, but only for grasping the meaning of a node in the context of the schema in which the node itself occurs. As an example,

[1]For a recent survey of existing methodologies, see the deliverable D2.2.3: *State of the art on current alignment techniques* at `http://knowledgeweb.semanticweb.org/`, which was produced as part of the EU funded Network of Excellence *KnowledgeWeb*. Perhaps it is worth knowing that there is a yearly initiative for comparing matching and alignment techniques called OAEI (Ontology Alignment Evaluation Initiative), whose goals are: assessing strength and weakness of alignment/matching systems, comparing performance of techniques, increase communication among algorithm developers, improve evaluation techniques, most of all, helping improving the work on ontology alignment/matching through the controlled experimental evaluation of the techniques performances see `http://oaei.ontologymatching.org/`.

imagine that the schema in in Figure 1 is used on a web site to classify multimedia assets and allow users to search pictures through navigation in the schema. Now consider the node n_9, which occurs at the end of the path

PICTURES/TRENTINO/COLOR/LAKES

As humans, we can guess that the documents classified under n_9 will be color picture of lakes in Trentino. However, this meaning is not directly accessible to machines, as it is only very partially encoded in the path itself; indeed, it is mostly "hidden" in the meaning of labels used to name the nodes, and in their arrangement in the path. In addition, we observe that our interpretation of the path heavily depends on a large amount of contextual and domain knowledge (e.g. that pictures can be in colors or black-and-white, that lakes have some geographical location, that Trentino is a geographical location, that pictures typically have a subject, and so on)[2].

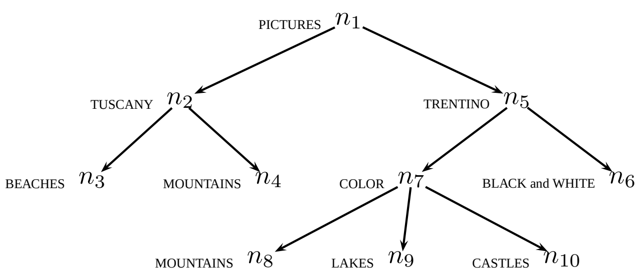

Figure 1: An example of classification schema

In (Bouquet et al., 2003), we identified at least three sources of knowledge which can be used to elicit the meaning of a node in a schema:

Lexical knowledge: knowledge about the words used in the labels. For example, the fact that the word 'picture' can be used in the sense of "a visual representation (of an object or scene or person or abstraction) produced on a surface" or in the sense of "a typical example of some state or quality" (sense #1 and #9 from WORDNET, respectively), and the fact that different words may have the same meaning (e.g., 'picture' and 'image');

Domain knowledge: knowledge about the meaning of labels, and about the relations between the meaning of different labels in the real world or in a specific domain. For example, the fact that Trentino is in Italy, or that pictures can be in color or back and wait;

Structural knowledge: knowledge deriving from the arrangement of nodes in the schema. For example, the fact that the set of objects classified at the end of the path PICTURES/TRENTINO/COLOR/LAKES is a subset of the objects that classified under the path PICTURES/TRENTINO/.

As we said, most matching methods focus on the third level. However, considering only structural knowledge is not enough, and may lead to at least two serious problems:

- we may be unable to discriminate between schemas that are structurally, but not semantically, isomorphic (for example, two paths like ANIMALS/MAMMALS/DOGS and PICTURES/TUSCANY/BEACHES are structurally isomorphic, but their semantically the first is a simple isA hierarchy, and the second is a complex concept like "pictures **of** beaches **located in** Tuscany";
- we may be unable to make any conjecture on the meaning of edges connecting nodes (elements) of a schema (see examples above, where meaning elicitation requires to understand that the relation between the nodes of the first path is "subclass of", whereas the relations between the nodes of the second path are "of" between pictures and beaches, and "located in" between the second and the third node.

This proves, in our opinion, that any attempt of designing a methodology for eliciting the meaning of schemas (basically, for reconstructing the intuitive meaning of any schema element into an explicit and formal representation of such a meaning) cannot be based exclusively on structural semantics, but must seriously take into account at least lexical and domain knowledge about the labels used in the schema[3]. In the next section we show how we did this.

3. WORDNET for semantic elicitation of schemas

In this section, we present a formal language called *Lexical Description Logic* (LDL) which in our approach we use to represent the meaning of elements in a classification schema.

LDL is the result of combining two main ingredients: a *logical language* (in this paper, use the logical language $\mathcal{ALCIO}$ which belongs to the family of *Description Logics*[4], and IDs of lexical entries in a lexicon – in the current implementation, IDs of WORDNET senses (in what follows, we denote by word#1, ..., word#n the n senses of the word "word". The main idea behind LDL is to use the DL language for representing structural meaning, and

[2]Similar arguments can be used for other structures, like Entity-Relationship schemas or RDF schemas. However, this paper mainly focuses on classifications, as they are quite well-understood also by non-specialists in computer science and Semantic Web.

[3]We say "at least", as there are other obvious types of knowledge which one may think of using, from an analysis of data associated with an element to more general contextual factors, like the application or the processes in which the schema is used. Here we ignore these other factors for the sake of simplicity, but of course they are relevant.

[4]Description logics are a family logical languages that are defined starting from a set of primitive concepts, relations and individuals, with a set of logical constructors, and has been proved to provide a good compromise between expressivity and computability. It is supported with efficient reasoning services (see for instance (Möller and Haarslev, 2003)).

any additional constraints (axioms) we might have from available domain knowledge; and to use WORDNET to anchor the meaning of labels in a schema to lexical meanings, which are listed and uniquely identified as WORDNET senses. Indeed, the primitives of any logical language do not have an "intended" meaning, and this is evident from the fact that, as in standard model-theoretic semantics, the primitive components of DL languages (i.e. concepts, roles, individuals) are interpreted, respectively, as generic sets, relations or individuals from some domain. What we need is a way for "grounding" their interpretation to the WORDNET sense that best approximates their intended meaning in a schema label. So, for example, a label like LAKES can be interpreted as a generic class in a standard DL semantics, but can be also assigned an intended meaning by attaching it to the the first sense in WORDNET (which in version 2.0 is defined as "a body of (usually fresh) water surrounded by land").

The advantage of LDL w.r.t. a standard DL encoding is that assigning an intended meaning to a label allows us to import automatically a body of (lexical) knowledge which is associated with a given meaning of a word used in a label. For example, from WORDNET we know that there is a relation between the class "lakes" and the class "bodies of water", which in turn is a subclass of physical entities. In addition, if an ontology is available where classes and roles are also lexicalized (an issue that here we do not address directly, but details can be found in (Serafini et al., 2004)), then we can also import and use additional domain knowledge about a given (sense of) a word, for example that lakes can be holiday destinations, that Trentino has plenty of lakes, even that a lake called "Lake Garda" is partially located in Trentino, and so on and so forth.

Technically, the idea described above is implemented by using WORDNET senses as primitives for a DL language. A LDL language is therefore defined as follows:

- the sets $\mathcal{C}$, $\mathcal{R}$ and $\mathcal{O}$ of (names for) primitive concepts, roles and individuals of LDL are subsets of WORDNET senses;
- complex concepts can be defined with the following production rule

$$
\begin{aligned}
C &:= \mathsf{c\#k} \mid C \sqcap C \mid C \sqcup C \mid \neg C \mid \forall R.C \mid \exists R.C \\
&\quad \mid \{\mathsf{o\#k}, \ldots, \mathsf{o\#k}\} \\
R &:= \mathsf{r\#k} \mid \mathsf{r\#k}^{-}
\end{aligned}
$$

  where $\mathsf{c\#k} \in \mathcal{C}$, $\mathsf{r\#k} \in \mathcal{R}$, and $\mathsf{o\#k} \in \mathcal{O}$;
- An axiom in LDL is an expression of the form $C \sqsubseteq D$, where C and D are complex concepts, and $C(a)$, where C is a concept and a is an individual.

The *formal semantics* of LDL is, as usual, a mathematical function $\cdot^{\mathcal{I}}$ that associates with each primitive concept C a set $C^{\mathcal{I}}$ of objects, to each primitive role R a binary relation $R^{\mathcal{I}}$, and to each individual o, an object $o^{\mathcal{I}}$. The formal semantics of complex concepts, and axioms can be defined inductively (see (Baader et al., 2003) for details).

The *intended semantics* of LDL is a new (derived) sense, which might not be in WORDNET, and that intuitively can be understood as a complex gloss obtained by combining the glosses of the lexical components. So, the intended semantics of $\mathsf{Car\#1} \sqcap \exists \mathsf{Color\#1}.\{\mathsf{Red\#1}\}$ is "a motor vehicle with four wheels; usually propelled by an internal combustion engine", which has "a visual attribute ... that results from the light it emits or transmit or reflect", which is "... the chromatic color resembling the hue of blood". In short, a red car.

As an example of the use of LDL descriptions for representing the meaning of a node in a schema, consider the node n_3 of the hierarchical classification of Figure 1. It would be represented as

$$\mathsf{image\#2} \sqcap \exists \mathsf{subject\#4}.\big(\mathsf{beaches\#1} \sqcap \exists \mathsf{Location\#1}.\big\{\mathsf{Tuscany\#1}\big\}\big)$$

and the intuitive semantics is "a visual representation produced on a surface of" "areas of sand sloping down to the water of a sea or lake" "situated in a particular spot or position" which is "a region in central Italy"

From this perspective, the problem of semantic elicitation can be rephrased as the problem of finding a LDL expression $\mu(n)$ for each element n of a schema, so that the intuitive semantics of μn is a good enough approximation of the intended meaning of the node.

3.1. Meaning Skeletons

Given a schema, the structural semantics associated with this schema provides the skeleton for the meaning of each node. Therefore our procedure will start from this skeleton, and will try to filter out "unlikely" skeletons by using extra, implicit semantics, obtained from lexical and domain knowledge.

Meaning skeletons are DL descriptions together with a set of axioms. The basic components of a meaning skeleton (i.e. the primitive concepts and roles) are the meanings of the single labels associated with nodes, denoted by $\lambda(n)$), and the semantic relations between different nodes (denoted by R_{ij}). Intuitively R_{ij} represents a semantic relation between the node n_i and the node n_j. As an example, we show how to construct the meaning skeleton of the hierarchical classification schema of Figure 1.

The intuitive structural semantics of a HC is that the meaning of a node is a specification of the meaning of its parent node. E.g., the intuitive meaning of a node labeled TRENTINO, with a parent node is IMAGES is "pictures of Trentino". In DL, this is encoded as $\mu(n) = \lambda(n) \sqcap \exists R_{nm}.\mu(m)$, where R_{nm} is some node that connets the meaning of n with that of m. If the label of n is for instance LAKELEVICO (a lake in Trentino) then the meaning of n is "pictures of Lake Levico in Trentino", then it is the meaning of the label of n that acts as modifier of the meaning of m. In description logics this is formalized as $\mu(n) = \mu(m) \sqcup \exists R_{mn}.\lambda(n)$. The choice between the first of the second case essentially depends both on lexical knowledge, which provides the meaning of the labels, and domain knowledge, which provides candidate relations between $\mu(m)$ and $\lambda(n)$. The following table summarizes some meaning skeletons associated with the HC of Figure 1:

node	meaning skeleton
n_1	$\lambda(n_1)$
n_2	$\lambda(n_1) \sqcap \exists R_{12}.\lambda(n_2)$ or $\lambda(n_2) \sqcap \exists R_{21}.\lambda(n_1)$
n_3	$\lambda(n_1) \sqcap \exists R_{12}.\lambda(n_2) \sqcap \exists R_{13}.\lambda(n_3)$ or $\lambda(n_2) \sqcap \exists R_{21}.\lambda(n_1) \sqcap \exists R_{13}.\lambda(n_3)$ or $\lambda(n_3) \sqcap \exists R_{31}.(\lambda(n_1) \sqcap \exists R_{12}.\lambda(n_2))$ or $\lambda(n_3) \sqcap \exists R_{31}.(\lambda(n_2) \sqcap \exists R_{21}.\lambda(n_1))$

Notice that, since at this level we do not have knowledge to distinguish which node is the modifier of the other, we must consider all the alternative meaning skeletons.

3.2. Local meaning ($\lambda(n)$)

The local meaning of a node in a schema, denoted by $\lambda(n)$, is a DL description approximating all possible meanings of the label associated with a node. To compute $\lambda(n)$, we make an essential use of WORDNET. If the label of a node n is a simple word like PICTURE, or FLORENCE, then $\lambda(n)$ represents all senses that this word can have in any possible context. For example, WORDNET provides 9 senses for the word 'picture' and 2 for 'Florence'. If m and n are nodes labeled with these two words, then $\lambda(m) =$ picture#1 $\sqcup$ picture#2 $\sqcup \ldots \sqcup$ picture#9 and $\lambda(n) =$ Florence#1 $\sqcup$ Florence#2.

When labels are more complex than a single word, as for instance "University of Trento", or "Component of Gastrointestinal Tract" (occurring in Galen Ontology (Rector et al., 1996)) then $\lambda(n)$ is a more complex DL description computable with advanced natural language techniques. The description of these techniques is beyond the scope of this paper and we refer the reader to (Magnini et al., 2003).

3.3. Relations between local meanings (R_{mn})

Domain knowledge (called a knowledge base) can be viewed as a set of facts describing the properties and the relations between the objects of some domain. For instance, a geographical knowledge base may contain the fact that Florence is a town located in Italy, and that Florence is also a town located in South Carolina. Clearly, the knowledge base will use two different constants to denote the two Florence. From this simple example, one can see how knowledge base relations are defined between meanings rather than between linguistic entities.

More formally, we define a knowledge base to be a pair $\langle T, A\rangle$ where T is a T-box (terminological box) and A is an A-box (assertional box) of some DL language. Moreover, to address the fact that knowledge is about meanings, we require that the atomic concepts, roles, and individuals that appear in the KB be taken from a set of senses provided by one (or more) linguistic resources. An fragment of knowledge base relevant to the examples given above is shown in Figure 2.

Domain knowledge is used to discover semantic relations holding between local meanings. Intuitively, given two primitive concepts C and D, we search the KB for a role R that possibly connect a C-object with a D-object. As an example, suppose we need to find a role that connects the concept picture#1 and the nominal concept {Florence#1)}; in the knowledge base of Figure 2, a candidate relation is

T-Box

picture#1 $\sqsubseteq \exists$Subject#4.physical object#1
picture#1 $\sqsubseteq \exists$size#1.size#2
city#1 $\sqsubseteq$ physical entity#1 $\sqcap \exists$located#2.country#4
picture#1 $\sqcap \exists$subject#4.(city#1 $\sqcap$ located#2.{Italy#2}) $\sqsubseteq$ picture#1 $\sqcap \exists$subject#4.{Italy#1}

A-Box

country#1(Italy#1)
country#1(South Carolina#1)
located#2(Florence#1, Italy#1)
located#2(Florence#2, South Carolina#1)

Figure 2: An example of Knowledge Base

subject#4. This is because Florence#1 is a *possible value* of the attributed Subject#4 of an Image#1.

More formally, R is a semantic relation between the concept C and D w.r.t., the knowledge base KB if and only if

i KB $\models C \sqsubseteq \exists R.E$ for some primitive concept E,

ii KB $\models D \sqsubseteq E$, and

iii for all primitive concepts F, KB $\models C \sqsubseteq \exists R.F$ implies that KB $\models E \sqsubseteq F$.

Conditions i–iii intuitively state that R is a role that connects C with D if, every C has an R which is F (condition i), and F is the smaller super-concept of D (conditions ii and iii) that has this property. By including R_{id} (the Identity Role $= \{x, x | x$ is any element of the domain$\}$) as a possible semantic relation between two concepts, the above definition captures the is-a relationship. Indeed, $C \sqsubseteq \exists R_{id} D$ is equivalent to $C \sqsubseteq D$ (C is a D). (Serafini et al., 2004) contains the detailed description of an algorithm for computing semantic relations between concepts.

According to this definition one can verify that Subject#4 is a semantic relation between Image#1 and the nominal concept {Florence#1}. Indeed KB $\models$ Image#1 $\sqsubseteq$ PhysicalEntity#1 (condition i), KB $\models$ {Florence#1} $\sqsubseteq$ PhysicalEntity#1 (condition ii) and for no other primitive concepts F different from PhysicalEntity#1 we have that KB $\models$ Image#1 $\sqsubseteq \exists$Subject#4.F (condition iii). Similarly Located#2 is a semantic relation between the nominal concepts Florence#1 and Italy#1, but it is not a semantic relation between Florence#2 and Italy#1.

The relations computed via conditions i–iii can be used also for disambiguation of local meanings. Namely, the existence of a semantic relation between two senses of two local meanings, constitutes an evidence that those senses are the right one. This allows us to discard all the others. For instance in the situation depicted in Figure 3, it to keep the sense Image#1 and eliminate the other two senses from the local meaning $\lambda(n_1)$. Similarly we prefer Florence#1 on Florence#2 since the former has more semantic relations that the latter.

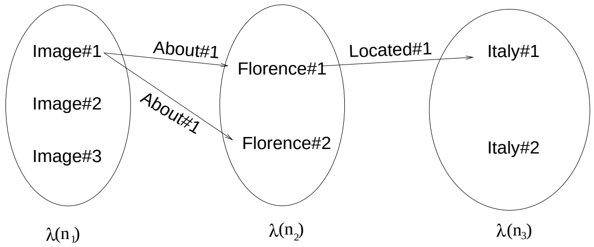

Figure 3: Semantic relation between senses

4. Matching HCs via semantic elicitation

In this section, we finally describe how our lexical-based method for meaning elicitation can be used to match hierarchical classifications (HCs).

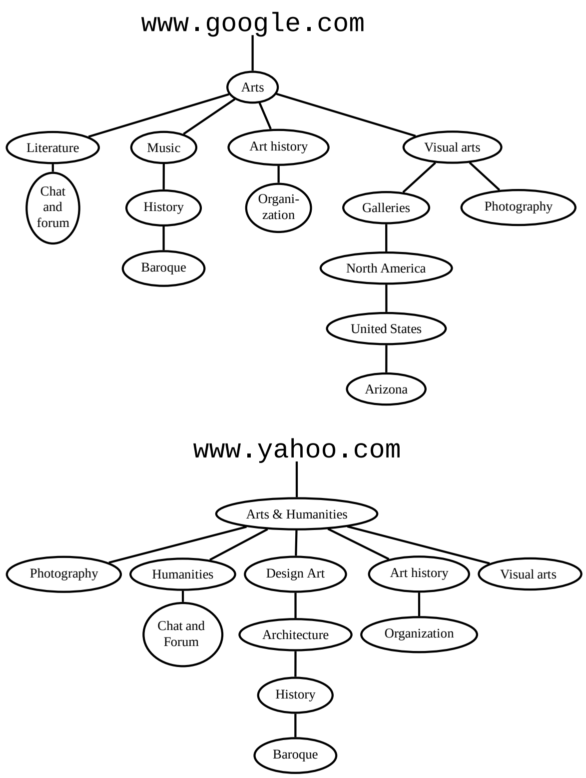

Figure 4: Two HCs on the Web

Consider the two HCs of Figure 4, which are real examples taken from the classification schemas of two well-known search engines. Imagine that a Web user is navigating the Google's directory, and finds an interesting category of documents (for example, the category named `Baroque` on the left hand side of the figure along the path `Arts > Music > History > Baroque`. She might want to find semantically related categories in other web directories. One way of achieving this result is by "comparing" the meaning of the selected category with the meaning of other categories in different directories. We show how this matching process as been implemented in CTXMATCH22.

Suppose that both Google and Yahoo had enabled their web directories with the semantic elicitation method described in the previous sections. This means that each node in the two web directories is equipped with a LDL formula which represents its meaning. In addition, we can imagine that each node contains also a body of domain knowledge which has been extracted from some ontology; this knowledge is basically what it is locally known about the content of the node (for example, given a node labeled `TUSCANY`, we can imagine that it can contain also the information that Tuscany is a region in Central Italy, whose capital is Florence, and so on).

Then, we can imagine that the following process is started:

1. the LDL formula representing the meaning of the node labeled `BAROQUE` in the Google's directory is extracted;

2. this formula is sent to Yahoo, together with a request to find semantically related nodes;

3. the semantic application at Yahoo tries to logically deduce (using any DL reasoner, like Racer, Pellet[5], or Fact[6]) whether any of the formulae attached to the local directory is in a relevant relation (e.g. equivalence, or subsumption) with the LDL formula attached to the request. Notice that, in computing potential relations, other background knowledge can be extracted from a local ontology to maximize the chances of discovering a relation;

4. if a Yahoo's LDL formula can be proved to be semantically related to the received formula, then the corresponding node in the web directory is returned as a search result. Otherwise, nothing is returned.

In the following table we present some relations automatically computed by CTXMATCH22 between the nodes of the two portion of Google and Yahoo in Figure 4.

Google node	Yahoo node	semantic relation
`Baroque`	`Baroque`	Disjoint $(\perp)$
`Visual Arts`	`Visual Arts`	More general than $(\supseteq)$
`Photography`	`Photography`	Equivalent $(\equiv)$
`Chat and Forum`	`Chat and Forum`	Less general than $(\subseteq)$

In the first example, CTXMATCH22 returns a 'disjoint' relation between the two nodes `Baroque`: the presence of two different ancestors (`Music` and `Architecture`) and the related world knowledge 'Music is disjoint with Architecture' allow us to derive the right semantic relation.

In the second example, CTXMATCH22 returns the 'more general than' relation between the nodes `Visual Arts`. This is a rather sophisticated result: indeed, world knowledge provides the information that 'photography IsA visual art' (`photography#1` $\rightarrow$ `visual art#1`). From structural knowledge, we can deduce that, while in the left structure the node `Visual Arts` denotes the whole concept (in fact `photography` is one of its children), in the right structure the node `Visual Arts` denotes the concept 'visual arts except photography' (in fact

[5]See `http://www.mindswap.org/2003/pellet/`.

[6]See `http://www.cs.man.ac.uk/~horrocks/FaCT`.

photography is one of its siblings). Given this information, it easy to deduce that, although despite the two nodes lie on the same path, they have different meanings.

The third example shows how the correct relation holding between nodes Photography is returned ('equivalence'), despite the presence of different paths, as world knowledge tells us that photography#1 → visual art#1.

Finally, between the nodes Chat and Forum a 'less general than' relation is found as world knowledge gives us the axiom 'literature is a humanities'.

5. Discussions

This is not the right context for an extensive discussion of the related work in the area of schema matching in the Semantic Web (we refer again to the deliverable D2.2.3 of KnowledgeWeb for this kind of discussion). We only say that, to the best of our knowledge, there is only another approach which uses similar ideas, called S-Match (), which is a refinement from the methodology underlying an earlier version of CTXMATCH2; no other approaches try to match schemas by eliciting the meaning of their elements in a systematic way, using in an integrated framework domain and lexical knowledge. What we'd like to do is to make a short list of open issues and problems which arose in our use of WORDNET for meaning elicitation.

A first comment is that in this paper, for the sake of simplicity, we assumed that all parties (Google and Yahoo in our example) use the same lexical resource for meaning elicitation, which basically means that schemas are annotated with expressions of the same LDL language. This makes everything easier, as two LDL formulae can be directly given to a reasoner for comparison. However, in (Bouquet et al., 2005), this assumption is relaxed, and it is discussed how *semantic peers* with different lexical resources and ontologies can still try to coordinate their local HCs.

Another comment concerns the difficulty we found is using WORDNET for annotating roles (relations). Indeed, a word sense like location#3 (as "a determination of the place where something is") can be both a concept and a role. This is the reason why, in the specification of the LDL language we described above, concepts, roles, and individuals, are not disjoint sets. Formally, this is not a problem, as the context where a primitive object occurs makes it possible to determine whether it must be considered a concept, a role, or an individual. It might be interesting to extend lexical resources with information about the "relational role" of some words.

Another, perhaps obvious comment, is that existing lexical resources become less and less useful when we move from generic to more specialized domains. Needless to say, the quality of our matching method decreases when we lack lexical information. The good news about the methodology we proposed is that, when it fails, it is always possible to know exactly why it failed. And the possible reasons are mainly three: (i) lack of lexical knowledge about a word; (ii) lack of domain knowledge about the meaning of a word and its relations with other meanings; (iii) wrong identification of relevant senses in the labels. All these problems can be fixed by incrementally extending the available lexical or domain knowledge.

Finally, we mentioned several times the need, in our approach, to "lexicalize" ontologies (KBs of domain knowledge) for integrating them into our framework. In our opinion, extending current ontology languages (like OWL) with lexicon-specific tags would be a very important improvement, as they would allow ontology engineers to make explicit the lexical meaning of the classes and properties they create, and thus provide a very valuable information to other applications which need to integrate or align independently developed ontologies.

6. Acknowledgments

The work described in this paper has been partly funded by the European Commission through grant to the project VIKEF (Virtual Information and Knowledge Environment Framework) under the number IST-507173.

7. References

Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

P. Bouquet, L. Serafini, and S. Zanobini. 2003. Semantic coordination: a new approach and an application. In K. Sycara, editor, *Second International Semantic Web Conference (ISWC-03)*, Lecture Notes in Computer Science (LNCS), Sanibel Island (Florida, USA), October.

Paolo Bouquet, Luciano Serafini, and Stefano Zanobini. 2005. Peer-to-peer semantic coordination. *Journal of Web Semantics*, 2(1).

C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press. ISBN 0-262-06197-X.

B. Magnini, L. Serafini, and M. Speranza. 2003. Making explicit the semantics hidden in schema models. In A. Cappelli and F. Turini, editors, *Proceedins of the 8th Congress of the Italian Association for Artificial Intelligence, Pisa, Italy*, number 2829 in Lecture Notes in Artificial Intelligence, pages 436–448. Springer Verlag.

R. Möller and V. Haarslev. 2003. Description logic systems. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The Description Logic Handbook*, chapter 8, pages 282–305. Cambridge University Press.

A. L. Rector, J. E. Rogers, and P. A. Pole. 1996. The galen high level ontology. In *Proceedings MIE 96*, pages 174–178. IOS Press, January.

L. Serafini, S. Sceffer, and S. Zanobini. 2004. Semantic coordination of hierarchical classifications with attributes. Technical Report T04-12-04, ITC-IRST.