Matching Hierarchical Classifications with Attributes

L. Serafini[2], S. Zanobini[1], S. Sceffer[2], and P. Bouquet[1]

[1] Dept. of Information and Communication Technology, University of Trento, via Sommarive, 14, 38050 Trento, Italy
[2] ITC–IRST, via Sommarive, 15, 38050 Trento, Italy
simsce@libero.it, serafini@itc.it, zanobini@dit.unitn.it

Abstract. Hierarchical Classifications with Attributes are tree-like structures used for organizing/classifying data. Due to the exponential growth and distribution of information across the network, and to the fact that such information is usually clustered by means of this kind of structures, we assist nowadays to an increasing interest in finding techniques to define *mappings* among such structures. In this paper, we propose a new algorithm for discovering mappings across hierarchical classifications, which faces the matching problem as a problem of deducing relations between sets of logical terms representing the meaning of hierarchical classification nodes.

1 Introduction

Hierarchical Classifications with attributes (HCAs) are tree-like structures with the explicit purpose of organizing/classifying some kind of data (such as documents, records in a database, goods, activities, services). Examples are: web directories (see e.g. the Google$^{\text{TM}}$ Directory or the Yahoo!$^{\text{TM}}$Directory), content management tools and portals, service registry, marketplaces, PC's file systems. Four very simple examples of such structures are depicted in Figure 1. In particular, consider the leftmost one: it has 4 nodes, labeled with the words `IMAGES`, `TUSCANY`, `BEACH`, `MOUNTAIN`. The nodes are connected by means of three edges, which are in turn labeled with 'subSet'. Finally, the node `IMAGES` is associated with an attribute $[\texttt{size} = \text{`}\texttt{large}\text{'}]$. As an example, the structure could be used for classifying the pictures taken during a vacation in Tuscany.

Due to the exponential growth and distribution of information across the network, and to the fact that such information is usually clustered by means of this kind of structures, we assist nowadays to an increasing interest in finding techniques to define *mappings* among such structures, namely a set of point-to-point relations between their nodes, in order to maximize the process of information retrieval. A lot of techniques for (semi-)automatically computing mappings have been proposed (see as an example [1, 2, 3, 4, 5]). Such methods associate to each pair of nodes occurring in two different HCAs a real number in [0,1], called *structural similarity*. As an example, consider the HCAs depicted in Figure 1-a: a matching technique could compute a structural similarity n between the two nodes `MOUNTAIN`.

Despite mappings are defined between HCA *nodes*, they obviously express relations between the *semantics* of HCA nodes. In our example, this means that the meanings of the two nodes `MOUNTAIN` – which we call *overall semantics* – are n–related. Our claim

Y. Sure and J. Domingue (Eds.): ESWC 2006, LNCS 4011, pp. 4–18, 2006.

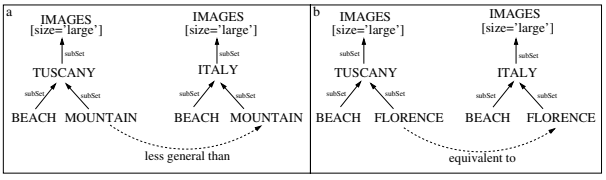

Fig. 1. Two simple pairs of Hierarchical Classifications with Attributes

is that such overall semantics of the nodes, which we can intuitively describe as 'Large size images of Mountains in Tuscany' and 'Large size images of Mountains in Italy' for left and right node respectively, is mainly implicit and it is the result of composing at least two further semantic (in this case, explicit) levels.

The first one, which we call *structural semantics*, is provided by the structure. This semantics says, as an example, that the node `TUSCANY` is a child of the node `IMAGES`, that the node `ITALY` has two children and so on, and that the node `IMAGES` is associated with an attribute [`size` = '`large`']. It further says that the arcs should be interpreted as 'subSet', as the set of documents that can be classified under the node `TUSCANY` is a subset of the set of documents that can be classified under the node `IMAGES`. We want to notice that the relation 'subSet' refers to the documents contained into the nodes and not to the concepts expressed by the labels, as no one would say that the concept 'Italy' is a subset of the concept 'Images'[1]. Anyway, this semantic level is not enough in order to decide for the right interpretation of the node: as an example, both 'large size images of Italy of mountains' (interpretation A) and 'large size images of mountains of Italy' (interpretation B) have the same probability to be the 'right' interpretation of the node `MOUNTAIN` of the most left hand HCA of Figure 1.

In order to decide for the right one, we need a further semantic level, which we call *external semantics*. This semantics, provided by the labels, relies on the knowledge associated to English words as 'Images', 'Tuscany', 'Florence' and 'Mountain'. We want to notice that such knowledge is shared by the community of English speakers and that is independent from their occurrence in the HCA. As an example, we know that the 'images' can depict physical objects, that the 'mountains' are physical objects and that 'Tuscany' is a spatial region where mountains occurs. Given this knowledge, we can decide for the interpretation 'large size images of mountains of Italy' (interpretation B) as the right one, and discard the other, 'large size images of Italy of mountains' (interpretation A), as wrong.

As the mappings express relations between the *overall semantics* of HCA nodes, and, in turn, such semantics strictly depends on these two semantic levels, a trivial consequence is that, in order to determine the 'right' relation holding between two nodes, we need to take into account both structural and external semantics. E.g., consider the two HCAs depicted in Figure 1-a. Intuitively, the relation between the nodes `MOUNTAIN` is 'less general than', as the documents contained into the leftmost node, intuitively 'large size images of Tuscan Mountains', are a subset of the images contained into the

[1] This distinction, in our opinion, is fundamental, as this is why this kind of structures cannot be considered *ontologies*. Indeed, an ontology describes the relations between the concepts, as the Hierarchical Classification describes the relations between sets of documents.

rightmost node, intuitively 'large size images of Italian Mountains'[2]. Consider now the two HCAs depicted in Figure 1-b. These HCAs are pairwise isomorphic to the ones depicted in Figure 1-a, as we simple substitute the label 'mountain' with the label 'Florence'. In this sense, these HCAs have (pairwise) the same structural semantics as the HCAs of Figure 1-a. But the relation between the nodes FLORENCE, corresponding to the nodes MOUNTAIN, is different, namely 'equivalent', as the documents contained into the leftmost node, intuitively 'large size images of Florence in Tuscany' are the same of the documents contained into the rightmost node, intuitively 'large size images of Florence in Italy'. In particular, the different relation is due to a different external semantics: Indeed, in this second case, external semantics provides that 'images' can depict physical objects, that 'Florence' is a physical object, and in particular a city, located both in 'Tuscany' and in 'Italy' (there is only one Florence in Italy and Tuscany).

The paper has the main goal of defining a procedure for matching Hierarchical Classifications with Attributes. In particular, the approach (i) makes explicit the (implicit) overall meaning of each HCA node in a Description Logics term, taking into account both structural and external semantics, and (ii) computes the relation between nodes comparing such DL terms. The paper goes as follows: Section 2 will formally define the matching problem for Hierarchical Classifications with Attributes, Section 3 will describe our approach, Section 4 will provide the algorithm description, and finally Section 5 will show the results of testing the algorithm on real examples.

2 The Problem

First of all, we introduce the notion of Hierarchical Classification with Attributes, whose four very trivial examples are depicted in Figure 1. In this paper we assume labels are English noun phrases. Let $\mathcal{N}$ be the set of such expression[3]. A HCA can be defined as follows:

Definition 1 (Hierarchical Classification with Attributes). *Let* $A = \{\langle a, b\rangle \mid a, b \in \mathcal{N}\}$ *be the set of all the possible pairs of strings in* $\mathcal{N}$ *(the set of attributes), where* $N = \{a \mid a \in \langle a, b\rangle, \langle a, b\rangle \in A\}$ *is the set of* attribute names *and* $V = \{b \mid b \in \langle a, b\rangle, \langle a, b\rangle \in A\}$ *is the set of* attribute values*. A* Hierarchical Classification with Attributes $G = \langle K, E, l_k, l_a\rangle$ *is a 4-tuple, where* K *is a set of nodes,* E *is a set of arcs and* $\langle K, E\rangle$ *is a rooted tree. Furthermore,* $l_k : K \rightarrow \mathcal{N}$ *is a function from nodes to* $\mathcal{N}$ *(the labels of the nodes), and* $l_a : K \rightarrow \{2^A \cup \emptyset\}$ *(the possibly empty set of node attributes).*

A Hierarchical Classification with Attributes can be intuitively described as a rooted tree where each node is associated with a natural language label and a (possibly empty) set of attributes. In this version of the algorithm we use a simplified definition of attribute as pairs name/value.

[2] When we say *the documents contained in some node*, we intend the documents *potentially* contained into the node. See [6] for a discussion on that.

[3] Examples of English noun phrases are single common words, as 'images' and 'Tuscany', complex words, as 'United States', expressions containing conjunctions, as 'big and small images'. This set is very difficult (perhaps impossible) to be formalized. In the following, we assume such a set as a primitive element.

Furthermore, let M be the set of all the possible overall meanings that the nodes of an HCA $G = \langle K, E, l_k, l_a \rangle$ can express, and let $\Upsilon : K \rightarrow M$ be a function that associates to each node of the HCA its overall meaning.

The definition of a *mapping*, namely a set of point-to-point relations between pairs of nodes of two distinct HCAs, represents the matching problem standard solution. As we defined a mapping as a set of relations between the overall semantics of the nodes, let $\Re = \{\sqsubseteq, \sqsupseteq, \equiv, \perp\}$ (where $\perp$ means 'disjoint') be the set of symbols expressing relations between meanings. A mapping can be formally defined as follows:

Definition 2 (Mapping). *A mapping $\mathcal{M}$ between two HCAs $G = \langle K, E, l_k, l_a \rangle$ and $G' = \langle K', E', l'_K, l'_a \rangle$ is a set of mapping elements $\langle m, n, R \rangle$ where m is a node in K, n is a node in K' and R is a semantic relation in $\Re$ holding between $\Upsilon(m)$ and $\Upsilon(n)$.*

3 Our Approach

A direct application of Definitions 1 and 2 suggest that a method for computing a mapping expressing relations between node overall semantics should be based, at least, on the following two steps:

Semantic Elicitation: the process for approximating the ideal function Υ;
Semantic Comparison: the process of finding mappings by comparing the Υ values.

As we have to handle machine–readable objects, we need to employ a concrete means to represent the range of the Υ function, i.e. the set of meanings. In this paper, we represent such meanings using Description Logic [7]. Let $S = \langle T, R \rangle$ be a signature for a DL language, where T is a set of primitive concepts and R is a set of primitive roles. Let $L = \langle \mathcal{C}, \mathcal{O} \rangle$ be a DL T–Box, where $\mathcal{C}$ is the set of concepts which can be defined by means of the signature S, and $\mathcal{O}$ a (possibly empty) set of axioms defined over $\mathcal{C}$. L represents the range of the function Υ, namely the set of all the overall meanings possibly expressed by HCA nodes.

In Section 1, we said that the overall meaning of a semantic graph node is the result of the composition of two different levels of semantics: external and internal. [8, 9] claim that the external semantics is given at least by the following knowledge sources:

Lexical knowledge: Such knowledge allows us to determine the (set of) concept(s) denoted by a word. E.g., the word 'Florence' can be used to denote at least two different concepts, namely 'a city in central Italy on the Arno' and 'a town in northeast South Carolina'. Conversely, it can be used to recognize that two different words may refer to the same concept. E.g. the words 'image' and 'picture' can denote the same concept 'a visual representation (of an object or scene or person or abstraction) produced on a surface', i.e. they are synonyms. Formally, let m be the set of words occurring in $\mathcal{N}$ (the set of meaningful expressions). A lexicon $\mathcal{L} : m \rightarrow 2^{T \cup R}$ is a function which associates each word to a set of *primitive concepts* or roles belonging to the signature of a T–Box L. Hereafter, we use Florence#1 to indicate the first concept possibly denoted by the word 'Florence'.

Ontological/World knowledge: This knowledge concerns the relations holding between primitive concepts. For example, the fact that there is a PartOf relation

between the concepts Florence#1, as 'a city in central Italy on the Arno', and Italy#1, as 'a republic in southern Europe on the Italian Peninsula', i.e. that 'Florence is part of Italy'. We formally define the ontological knowledge as the set of axioms of a T–Box L (namely $\mathcal{O}$).

On the other hand, the internal semantics is provided by node arrangement into the HCA. E.g., consider the node FLORENCE of the rightmost graph of Figure 1. Since the HCA is a tree, the internal semantics of the node is represented by the fact that the node IMAGES is the root and that the node FLORENCE lies in the path IMAGES/ITALY/FLORENCE. Furthermore, it says that an attribute [size = 'large'] is associated with the node IMAGES. Finally, it says that the node FLORENCE (possibly) contains a subset of the documents (possibly) contained in the node ITALY, which in turn (possibly) contains a subset of the documents (possibly) contained in the node IMAGES.

During the semantic elicitation step, the two semantic levels will be combined, as shown in Section 4.1, in order to obtain the overall meaning of the nodes. Going on with our example, the intuitive meaning of the node FLORENCE, 'large size images of Florence in Italy', will be approximated with the following DL term:

$$\text{Image\#2} \sqcap \exists \text{size\#1.large\#1} \sqcap \exists \text{about\#1.}(\text{Florence\#1} \sqcap \exists \text{PartOf.Italy\#1}) \quad (1)$$

where Image#2 represents the concept 'a visual representation (of an object or scene or person or abstraction) produced on a surface', size#1 the concept 'the physical magnitude of something (how big it is)', Florence#1 the concept 'a city in central Italy on the Arno', and so on.

Finally, the semantic comparison step (Section 4.2) determines the relation holding between two nodes by comparing their meanings (formalized during the semantic elicitation step). For this task, we rely on existent techniques for determining possible entailment between concepts. As an example, imagine that during the semantic elicitation step we determine the following overall semantics for the node FLORENCE of rightmost HCA of Figure 1-b:

$$\text{Image\#2} \sqcap \exists \text{size\#1.large\#1} \sqcap \exists \text{about\#1.}(\text{Florence\#1} \sqcap \exists \text{PartOf.Tuscany\#1}) \quad (2)$$

During the semantic comparison step, we check if it holds that $\mathcal{O} \models (1) \equiv (2)$[4]. From this fact, we can conclude that the relation between the nodes FLORENCE of HCAs of Figure 1 is 'equivalent'.

4 Algorithm Description

In this section, we describe a possible implementation of the two main steps described in the previous section. In this implementation, the lexicon $\mathcal{L}$ is represented by WORDNET[5]. Both terms T and roles R of the signature S are the synsets of WORDNET. R

[4] Of course, this is true only if the ontology provides that Florence#1 $\sqsubseteq \exists$PartOf.Tuscany#1 and Florence#1 $\sqsubseteq \exists$PartOf.Italy#1.

[5] WORDNET [10] is a well-known lexical/ontological repository containing the set of concepts denoted by words (called synsets, i.e. set of synonyms), and a small set of relations (e.g. IsA and PartOf) holding between senses.

contains also two predefined roles: IsA and PartOf. The set of concepts $\mathcal{C}$ is the set of all the allowed expressions built on signature S. The ontology $\mathcal{O}$ is composed both by the IsA and PartOf relations defined in WORDNET and by a further *ad hoc* ontology.

Furthermore, we define *focus* of a node n the part of the structural semantics which we take into account in order to build the overall meaning of n[6]. Formally:

Definition 3 (Focus of a node). *The* focus *of a node n of a HCA G is the HCA $F(n) = \langle K', E', l'_k, l'_a \rangle$, such that: (i) K' contains the nodes of the path from the root of G to n; (ii) for each $k \in K'$, K' contains all its attributes too; (iii) all the other elements of $F(n)$ are the restriction of the corresponding component of G on K'.*

Now we can describe the implementation of the two main steps of the algorithm, namely *semantic elicitation* and *semantic comparison*.

4.1 Semantic Elicitation

The semantic elicitation process has the main aim of approximating the meaning of each node of a HCA, namely it is an implementation of the function Υ. If applied to the node FLORENCE of the rightmost HCA of Figure 1-b, it should generate a representation of the intuitive meaning 'large size images of Florence in Italy'. In particular, we apply the following three sub-steps: (i) we build the node *local meaning*, i.e. the meaning of the node taken in isolation (intuitively represented, for nodes IMAGES, ITALY and FLORENCE by 'large size images', 'Italy' and 'Florence'), (ii) we discover possible relations between the local meanings (e.g. a possible relation between the meanings of 'large size images' and 'florence' is that the images are 'about' Florence), and (iii) we combine them, in order to obtain the *global meaning*, namely the meaning of the node in the HCA (intuitively represented by 'large size images about Florence in Italy'). In the following sections we provide a description of them, and an example on how they operate on the node FLORENCE of the rightmost HCA of Figure 1-b.

1. Building the local meaning. In this phase, we consider separately each node of an HCA and, for each of them (with its set of attributes), we generate a DL description in $\mathcal{C}$ which approximates all possible meanings of the node. Imagine that we consider a labeled node with n attributes: The local interpretation of the node is generated starting from the following pattern: $\mathsf{label} \sqcap \exists \mathsf{attName}_1.\mathsf{filler}_1 \sqcap \ldots \sqcap \exists \mathsf{attName}_n.\mathsf{filler}_n$, where label is the label of the node, and $\mathsf{attName}_j$ and filler_j are the attribute name and the filler of the j^{th} attribute respectively. In particular, we consider the attribute name as a role and the attribute filler as a range of a DL term. We obtain the space of all the possible interpretations of a node by substituting the words occurring in the pattern (namely the labels of the node, and of the attributes) with each concept possibly denoted by the words themselves w.r.t. lexicon $\mathcal{L}$. Of course, a label, an attribute name or a filler can contain a word not present in the lexicon. In this case, we consider the string itself as a concept. As an example, if we find the word 'frtye' in a label, the resulting concept will

[6] Other possible definitions can be provided. In [8] we define focus of a node n the set of nodes occurring in the path from root to n, and their respective children. In the extreme case, we can consider all the nodes of the HCA.

be frtye#1. Obviously no relation of synonymy will be found for this concept, except for concepts denoted by the same string. Furthermore, the current implementation can handle also more complicated cases than single words. In particular, any English noun phrase occurring in the label, in the attribute name or in the filler can be treated. In order to do that, we use a dedicated natural language parser for individuating the syntactic category and the part of speech of the words. The description of such parser is out of the scope of this paper. Details can be found in [11]. Going on our example, as our lexicon provides 7 concepts for the word 'Images', 5 for 'size', 8 for 'large', 1 for 'Italy' and 2 for 'Florence', the space of all the possible interpretations for the nodes IMAGES, ITALY and FLORENCE of rightmost schema of Figure 1-b is the following:

Table 1. Set of all the possible *local interpretations*

i(IMAGES)	i(ITALY)	i(FLORENCE)
Image#1 $\sqcap$ $\exists$size#1.large#1	Italy#1	Florence#1
Image#2 $\sqcap$ $\exists$size#1.large#1		Florence#2
...		
Image#7 $\sqcap$ $\exists$size#5.large#8		

Of course, not all the concepts denoted by words have to be considered in order to compose the node meaning, as some role or concept couldn't be the *right* one w.r.t. *that* node in *that* HCA. E.g., the concept Florence#2 ('a town in northeast South Carolina'), possibly denoted by the word 'Florence', is probably wrong in order to represent the meaning of the node FLORENCE (as the Florence we are talking about seems to be the Florence in Italy). The next phase has the aim of discarding such useless concepts.

First of all, we try to discover semantic relations holding between the concepts associated to nodes, accessing ontology $\mathcal{O}$. In the following, we assume to have a black box function $\mathcal{R} : T \times T \rightarrow R$ which takes as input two concepts and returns the role holding between them[7]. In particular, we search for the relations tieing two different kinds of elements:

Attribute Roles: Consider the node IMAGES, where an attribute occurs. In the previous phase, we build the set of all the possible interpretations for this node (i(IMAGES)). In this step, we access ontology $\mathcal{O}$ for determining if it *explicitly supports* one (or more) of these possible interpretations. As an example, we can discover that $\mathcal{R}$(Image#2, large#1) = size#1, namely that the second interpretation of Table 1 is supported by the ontology.

Structural Roles: In this step we search for semantic relations relating concepts belonging to different nodes. In particular, as a focus represents the set of nodes to take into account in order to build the meaning of a node, we search for relations holding between concepts of nodes occurring in the same focus. As an example,

[7] The description of this methodology for extracting relations between concepts is out of the scope of this paper. A detailed description can be founded in [11]. As an example of how this procedure works, imagine that the ontology $\mathcal{O}$ contains the following axioms: Image#2 $\sqsubseteq$ $\exists$about#1.Entity#1 and Florence#1 $\sqsubseteq$ Entity#1. Then, $\mathcal{R}$(Image#2, Florence#1) = about#1.

consider node FLORENCE. As the node IMAGES occurs in the focus of FLORENCE, we search for a relation holding between the concepts denoted by the word 'Image' and the concepts denoted by the word 'Florence'. In this case we find the following relation: $\mathcal{R}(\mathsf{Image\#2}, \mathsf{Florence\#1}) = \mathsf{about\#1}$.

Table 2 shows the set of relations we find w.r.t. the focus of node FLORENCE. For the relations we use the notation $\langle \mathsf{Image\#2}, \mathsf{large\#1}, \mathsf{size\#1} \rangle$ for indicating the relation $\mathcal{R}(\mathsf{Image\#2}, \mathsf{large\#1}) = \mathsf{size\#1}$ discovered by function $\mathcal{R}$.

Table 2. Set of relations

1	$\langle$Image#2, large#1, size#1$\rangle$
2	$\langle$Image#2, Italy#1, about#1$\rangle$
3	$\langle$Image#2, Florence#1, about#1$\rangle$
4	$\langle$Image#2, Florence#2, about#1$\rangle$
5	$\langle$Florence#1, Italy#1, PartOf$\rangle$

Then, filtering step is performed by applying the following rules to each concept extracted in the previous phases.

Weak rule: A concept c associated to a word w occurring in a node n can be removed if c it is *not* involved in any relation and exists another concept c' (different from c) associated to w in n which is involved in some relation.

Strong rule: A concept c associated to a word w occurring in a node n can be removed if c it is *not* involved in any IsA or PartOf relation and exists another concept c' (different from c) associated to w in n which is involved in some IsA or PartOf relation.

An example of the application of the first rule is the following. In Table 2 we find that the concept Image#2 is involved in relations 1–4, while the concepts Image#1, Image#3, ..., Image#7 are not involved in any relation. From this fact, we can guess that the 'right' concept expressed by the word 'Image' in this node is Image#2, and the other ones can be discarded. The second rule says something stronger, as a concept can be discarded even if involved in some relation. The idea is that we consider IsA and PartOf relations be strongest than the other ones. As an example, consider the relations 4 and 5. Because of we find a relation PartOf between the concepts Florence#1 and the concept Italy#1 (relation 5), we can discard the concept Florence#2 also if involved in a about#1 relation (axiom 4). The consequence of this step is to reduce the space of possible interpretations of a node, discarding any interpretation involving discarded concepts. The concepts are considered accessing top-down the hierarchy. Obviously, as these are heuristic rules, mistakes can be performed. Following Table shows the current interpretations for the nodes IMAGES, ITALY and FLORENCE[8].

[8] Note that a node can have more than one possible interpretation. When this happens, all the interpretations are kept (ambiguity partially solved). Formally, an interpretation of a node n with more than one possible interpretation is encoded as the disjunction ($\sqcup$) of all the possible interpretations occurring in the $i(n)$ set.

$i(\texttt{IMAGES})$	$i(\texttt{ITALY})$	$i(\texttt{FLORENCE})$
Image#2 $\sqcap \exists$size#1.large#1	Italy#1	Florence#1

2. Determining the relations between nodes. In this phase, we try to find relations tieing different nodes. To this end, we take into account the subset of the previously extracted relations (see Table 2), and in particular the set of relations holding between concepts belonging to different nodes. Going on our example, we take into account relations 2–5.

As for concepts, not all the relations are the right ones for expressing the node meanings. In order to individuate the 'right' ones, first of all we discard the relations involving discarded concepts, as they refer to no longer existent concepts. The following Table shows the current set of relations.

1	$\langle$Image#2, large#1, size#1$\rangle$
2	$\langle$Image#2, Italy#1, about#1$\rangle$
3	$\langle$Image#2, Florence#1, about#1$\rangle$
5	$\langle$Florence#1, Italy#1, PartOf$\rangle$

Then we cluster this set in homogeneous triples $\langle \texttt{M}, \texttt{N}, r\rangle$, where M and N are nodes and r is a role in R. In particular, a relation $\langle$C#j, D#k, R#t$\rangle$ belongs to the triple $\langle \texttt{M}, \texttt{N}, r\rangle$ if C#j is present in $I(\texttt{M})$ (the local interpretation of M), D#k is present in $I(\texttt{N})$ (the local interpretation of N) and R#t $= r$. Let T be such a set of triples: it represents the *relations between nodes*, that we call *edges*. As the meaning of a node n is determined by its focus, in order to build its meaning, we need to take into account the set of edges relating nodes occurring in the focus. Concerning the node FLORENCE, the edges we are interested in are depicted in the following Table:

1	$\langle$IMAGES, ITALY, about#1$\rangle$
2	$\langle$IMAGES, FLORENCE, about#1$\rangle$
3	$\langle$FLORENCE, ITALY, PartOf$\rangle$

Of course, such set can be ambiguous. E.g., may happen that we have two different edges between the same pair of nodes, or that two nodes are mutually in relation, and so on. Formally, let n be a node, F its focus, T_F the set of edges restricted to F and $G = \langle F, T_F\rangle$ the graph obtained by combining the set F of nodes and the set T_F of edges. We define G *not ambiguous* if (i) it is acyclic, (ii) don't exist two edges between the same two nodes and (iii) each node has at most one entering edge. In our example, G results ambiguous: indeed, node ITALY has two entering edges: from IMAGES (edge 1) and from FLORENCE (edge 3)[9]. Let $\Gamma \subseteq 2^G$ be the set of all the possible unambiguous sets of graphs. As the presence of edges 1 and 3 creates the ambiguity, Γ contains the graphs $\langle F, \langle 1, 2\rangle\rangle$, $\langle F, \langle 2, 3\rangle\rangle$, $\langle F, \langle 1\rangle\rangle$, $\langle F, \langle 2\rangle\rangle$, $\langle F, \emptyset\rangle$. In order to choose the graph

[9] Intuitively, it means that the node ITALY, and consequently the concept it denotes, intuitively 'Italy', cannot be considered a modifier of two different concepts, namely 'Florence' (Florence is part of Italy) and 'Images' (images about Italy). Intuitively, we need to decide for the ambiguity between 'images about Italy in Florence' and 'images about Florence in Italy'.

representing the set of 'right' relations between nodes, we apply the two following heuristics rules:

Edges Maximization: Such rule prefers graph(s) with larger set of edges. Intuitively, it selects the set which better connect the nodes in the HCA. In our example, the rule prefers the graphs $\langle F, \langle 1, 2\rangle\rangle$ and $\langle F, \langle 2, 3\rangle\rangle$ of Γ;

Path Minimization: Such rule prefers graph(s) minimizing the number of paths representing a node coverage over F. As an example, the rule prefers the graph $\langle F, \langle 2, 3\rangle\rangle$ of Γ, as can be defined a node coverage using the single path ($\texttt{IMAGES} \xrightarrow{\text{about\#1}} \texttt{FLORENCE} \xrightarrow{\text{PartOf}} \texttt{ITALY}$)[10].

By applying these rules, we can determine that the graph representing the right set of relations between the nodes occurring in the focus of the node FLORENCE is $\langle F, \langle 2, 3\rangle\rangle$[11].

3. Composing the meaning. In the last step, local meanings are combined by means of the edges in order to obtain the *global meaning* of a node. Formally, it can be defined as follows:

Definition 4 (Global Meaning). *Let n be a node in a HCA, F the focus of n and $G = \langle F, T\rangle$ a unambiguous graph. Furthermore, let Π the minimal set of paths representing a coverage of G. The global meaning $g(n)$ for the node n is the following DL term:*

$$\bigsqcap_{\pi \in \Pi} I(\pi_0) \sqcap \phi(R^{\pi_0,\pi_1})(I(\pi_1) \sqcap \phi(R^{\pi_1,\pi_2})(I(\pi_2) \sqcap \ldots)\ldots)$$

where $I(\texttt{Y})$ is the local meaning of the node Y, π_j is the j^{th} elements of the path π (a node of the HCA), R^{π_k,π_j} is the relation between the k^{th} and j^{th} node in π provided by T and $\phi(r)$ is $\emptyset$ if $r = \mathsf{IsA}$, '$\exists r.$' otherwise.

Essentially, we define the global meaning of a node as the conjunction ($\sqcap$) of all the meanings recursively build on the paths of the coverage. In our example, where G contains the single coverage:

$$\texttt{IMAGES} \xrightarrow{\text{about\#1}} \texttt{FLORENCE} \xrightarrow{\text{PartOf}} \texttt{ITALY}$$

the following term (3) represents the *global meaning* for node FLORENCE:

$$\mathsf{Image\#2} \sqcap \exists \mathsf{size\#1.large\#1} \sqcap \exists \mathsf{about\#1.(Florence\#1} \sqcap \exists \mathsf{PartOf.Italy\#1)} \tag{3}$$

4.2 Semantic Comparison

The second macro–step of the algorithm consists in computing the relations holding between nodes comparing their meanings via logical reasoning. In this phase we exploit well–known techniques[12], and in particular we use RACER DL reasoner (see *http://www.sts.tu-harburg.de/~r.f.moeller/racer*).

[10] As an example, the other graph, namely $\langle F, \langle 1, 2\rangle\rangle$, should be represented by the following two paths: $\texttt{IMAGES} \xrightarrow{\text{about\#1}} \texttt{FLORENCE}$ and $\texttt{IMAGES} \xrightarrow{\text{about\#1}} \texttt{ITALY}$.

[11] When ambiguity arises anyway, we perform a random choice.

[12] Reasoning complexity is directly related to the Description Logic degree we use for encoding the meaning. In this paper we use the ALC fragment of DL, which guarantees decidability.

Imagine semantic elicitation step has been completed for each node occurring in the two different HCAs $H = \langle K, E, l_k, l_a \rangle$ and $H' = \langle K', E', l'_k, l'_a \rangle$. Then we perform the following reasoning problems for each pair of nodes $k \in K$ and $k' \in K'$ (remember that the function $g()$ returns the DL terms approximating the node meanings):

Entailment problem	Semantic Relation
$\mathcal{O} \models (g(k) \sqcap g(k')) \sqsubseteq \bot$	$\bot$
$\mathcal{O} \models g(k) \equiv g(k')$	$\equiv$
$\mathcal{O} \models g(k) \sqsubseteq g(k')$	$\sqsubseteq$
$\mathcal{O} \models g(k) \sqsupseteq g(k')$	$\sqsupseteq$
otherwise	$*$

Ontological knowledge $\mathcal{O}$ can be used in order to improve reasoning process. In case no relation is founded, we return the generic relation $*$, that we interpret as *possible intersection* or *compatibility*.

E.g, suppose that we want to find the relation holding between the nodes FLORENCE of leftmost and rightmost HCA of Figure 1-b respectively. The semantic elicitation step produces the description (3) for node FLORENCE of the rightmost HCA of Figure 1-b. Now, imagine that we apply the same process to the node FLORENCE of the leftmost HCA of Figure 1-b. We obtain the following DL term:

$$\mathsf{Image\#2} \sqcap \exists \mathsf{size\#1.large\#1} \sqcap \exists \mathsf{about\#1.(Florence\#1} \sqcap \exists \mathsf{PartOf.Tuscany\#1)} \quad (4)$$

Moreover, imagine that the ontology $\mathcal{O}$ provides the following axioms[13]: $\mathsf{Florence\#1} \sqsubseteq \exists \mathsf{PartOf.Italy\#1}$ (Florence is part of Italy), $\mathsf{Florence\#1} \sqsubseteq \exists \mathsf{PartOf.Tuscany\#1}$ (Florence is part of Tuscany) and $\mathsf{Tuscany\#1} \sqsubseteq \exists \mathsf{PartOf.Italy\#1}$ (Tuscany is part of Italy). We can easily state that $\mathcal{O} \models (3) \equiv (4)$ holds. So, we can conclude that the *semantic relation* between the nodes FLORENCE of leftmost and rightmost HCAs of Figure 1-b respectively is 'equivalent' ('$\equiv$').

We want to notice that the same process we describe, when applied to the nodes MOUNTAIN of Figure 1-a, gives different results. Indeed, consider the following global meanings associated to the nodes MOUNTAIN of leftmost and rightmost HCA respectively:

$$\mathsf{Image\#2} \sqcap \exists \mathsf{size\#1.large\#1} \sqcap \exists \mathsf{about\#1.(Mountain\#1} \sqcap \exists \mathsf{located\#1.Tuscany\#1)} \quad (5)$$

$$\mathsf{Image\#2} \sqcap \exists \mathsf{size\#1.large\#1} \sqcap \exists \mathsf{about\#1.(Mountain\#1} \sqcap \exists \mathsf{located\#1.Italy\#1)} \quad (6)$$

As before, imagine that the ontology $\mathcal{O}$ provides the following axioms: $\mathsf{Florence\#1} \sqsubseteq \exists \mathsf{PartOf.Italy\#1}$, $\mathsf{Florence\#1} \sqsubseteq \exists \mathsf{PartOf.Tuscany\#1}$ $\mathsf{Tuscany\#1} \sqsubseteq \exists \mathsf{PartOf.Italy\#1}$ We can state that $\mathcal{O} \models (5) \sqsubseteq (6)$ holds[14]. So, we can conclude that the *semantic relation* between the nodes MOUNTAIN of leftmost and rightmost HCAs of Figure 1-a respectively is 'less general than' ('$\sqsubseteq$').

[13] All these axioms are derived from WORDNET.

[14] The conclusion strictly doesn't hold. In the current version of the algorithm, we use the following meta-rule: if $\mathsf{A} \sqsubseteq \exists \mathsf{PartOf.B}$ then $\mathsf{A} \sqsubseteq \mathsf{B}$, namely that we treat the 'part of' relation as an 'is a' relation.

5 Testing the Algorithm

The algorithm have been intensively tested on two tasks, driven from the 2^{nd} international ontology alignment competition[15]. The first task consists in trying to align a bibliography ontology against many others ontologies, some related, some others not. The second task consists on aligning subsets of the three biggest web directories available on line, namely Google, Yahoo and Looksmart. We use both as lexical and ontological knowledge WORDNET. No *ad hoc* ontology has been used.

Before introducing the results, we have to formally define the notion of correct result. Let $\mathtt{A}$ and $\mathtt{B}$ two nodes in two different HCs, and $\langle \mathtt{A}, \mathtt{B}, R \rangle$ the mapping determined by the algorithm. This is *strongly* correct if in the Golden standard is present a mapping $\langle \mathtt{A}', \mathtt{B}', R' \rangle$ such that $A = A'$, $B = B'$ and $R = R'$, and is *weakly* correct if $A = A'$, $B = B'$ and $R' \rightarrow R$[16]. As measure of accuracy for the algorithm results, we use the standard *Precision*, *Recall* and *F-Measure*[17].

Matching The Benchmark. The task consists in aligning a *reference ontology* against 51 other ontologies (hereafter *target ontologies*). The reference ontology is based on one of the first EON Ontology Alignment Contest, and it has been improved by comprising a number of circular relations that were missing from the previous test. The domain of the reference ontology is the Bibliographic references, and represents a subjective view of what must be a bibliographic ontology. It contains 33 named classes. The target ontologies represent a sort of alteration of the reference ontology. In particular, there are 5 categories of alteration: (i) names of entities can be replaced by random strings, synonyms, names with different conventions, strings in another language than English; (ii) comments can be suppressed or translated in another language; (iii) specialization hierarchy can be suppressed, expansed or flattened; (v) classes can be expanded, (namely replaced by several classes) or flattened[18]. On the whole, the target ontologies contains 2,044 named classes. Beside the reference and the target ontologies, a golden standard have been proposed. The algorithm performs all the comparisons in 10 minutes, using a common laptop, namely a *Toshiba A60-122* with the following characteristics: CPU Intel Pentium 4 3.06 GHz, 704 MB RAM, HD 30 GB (4500 RPM), OS Microsoft Windows XP SP2. The following Table reports Average Precision, Recall and F-Measure:

	Precision	**Recall**	**F-Measure**
Strong	0.48	0.48	0.47
Weak	0.66	0.64	0.64

[15] The testing results can be obtained at `http://www.stefanozanobini.net/`.

[16] That is the Golden Standard result must imply the algorithm result. As an example, if the relation present in the Golden standard is $\equiv$, than both $\sqsubseteq$ or $\sqsupseteq$ relations are weakly correct, whereas if the relation present in the Golden standard is $\sqsubseteq$, than $\sqsupseteq$ relation is incorrect. Note that both these conditions of correctness are stronger than the standard one, which states that a mapping $\langle \mathtt{A}, \mathtt{B}, R \rangle$ is correct if in the Golden standard is present a mapping $\langle \mathtt{A}', \mathtt{B}', R' \rangle$ such that $A = A'$, $B = B'$.

[17] The F-measure is usually defined as $F = 2 * \frac{(Recall \times Precision)}{(Recall + Precision)}$.

[18] See `http://oaei.inrialpes.fr/2005/` for a detailed description of the alterations.

Matching Web Directories. The task consists in evaluating the performances of the algorithm on matching real world HCs. The evaluation dataset has been extracted from Google, Yahoo and Looksmart web directories, and consists in 2,265 sub-tasks, where each sub-task is represented by a pair of subsets of the mentioned web directories. Each single subset is represented as an OWL ontology, where classification relations are modeled as OWL *subClassOf* relations. In the following, we call *source* and *target* the first and the second ontology of each sub-task respectively. The set of sources has on the whole 14,845 concepts, while the set of targets has on the whole 20,066 concepts. The algorithm performs all the comparison in 18 minutes, using the same machine as the previous task. As no golden standard has been provided, the algorithm accuracy has been manually verified. Due to time reasons, only a random 3% of the sub-tasks have been verified. The following Table reports Precision, Recall and F-Measure:

	Precision	**Recall**	**F-Measure**
Strong	0.65	0.55	0.55
Weak	0.72	0.62	0.61

6 Related Work

The algorithm faces the problem of matching HCs deducing relations between logical terms approximating the meaning of the nodes. Under this respect, to the best of our knowledge, there are no other works to which we can compare ours. Standard graph matching techniques (for a survey, see [12]), essentially rely on finding isomorphisms between graphs or sub-graphs. Of course, as in real world applications graph representations present an high degree of heterogeneity, we can't expect to individuate a perfect match between them. So, methods for computing similarity between pairs of elements have been proposed (see as an example [1, 2, 3, 4, 5]). In particular, such methods compute structural similarity, usually expressed by a real number in [0,1], across all the pairs of nodes occurring in different graphs. Essentially, such structural similarity takes into account only on what we call *structural semantics*, and, essentially, such methods present the drawbacks we show in Section 1.

Recently, approaches which combines graph matching techniques with lexical knowledge have been proposed. The most relevant, in our opinion, are CUPID [13], a completely automatic algorithm for schema matching, and MOMIS [14], a set of semi–automatic tools for information integration of (semi)structured data sources. Both approaches exploit lexical information in order to increase the node similarity. But, as they essentially rely on standard graph matching techniques, taking into account only *structural semantics*, they also partially present the same drawbacks. As an example, in case of equivalent nodes occurring in completely different structures, and completely unrelated nodes that belong to isomorphic structures, the matches fail.

On the other hand, the task of translating some natural language expression into a formal expression is partially shared with the NL community. As an example, [15] proposes a method for building schemas starting from NL statements. The problem is essentially the other way round with respect to the ours, with the main difference that rely essentially on syntactical (grammatical) structure, and not on semantic relations, as we do. [16] proposes a method for interpreting schema elements (with particular

emphasis on Web directories). But they essentially analyzes the documents contained on the schemas, as we analyzes the schema.

The problem of individuating the right concept expressed by some word is commonly known as *word sense disambiguation* (see [17] for a survey). We face a similar problem in the filtering step. The heuristics we use are essentially different from the most of that procedures, as they are based on the notion of conceptual distance among concepts (see [18] and [19]), which in turn rely on standard graph matching techniques.

7 Conclusions

In this paper we presented a new approach for matching hierarchical classifications with attributes. The algorithm is essentially an improvement of that one presented in [8], and of its extension in [20]. In particular, we extend the algorithm in order to (i) treat node attributes, (ii) encode the meaning using a more powerful language (Description Logics vs Propositional Logics), and (iii) allow complicate node labels (Noun Phrases expressions vs simple words).

We want to stress that the main algorithm novelty consists in not considering a HC as a (semantically) homogeneous structure. Indeed, its semantic is the result of merging at least two further different semantic levels: *structural semantics* (represented by the backbone of the structure and by the interpretation of the arcs) and *external semantics* (represented by the interpretations of the labels and by the relations between the concepts).

Finally, we want to notice that the approach we describe can be 'easily' extended in order to treat, in principle, any graph-like domain representation.

References

1. Zhang, K., Wang, J.T.L., Shasha, D.: On the editing distance between undirected acyclic graphs and related problems. In Galil, Z., Ukkonen, E., eds.: Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching. Volume 937., Espoo, Finland, Springer-Verlag, Berlin (1995) 395–407
2. Pelillo, M., Siddiqi, K., Zucker, S.W.: Matching hierarchical structures using association graphs. Lecture Notes in Computer Science **1407** (1998)
3. Milo, T., Zohar, S.: Using schema matching to simplify heterogeneous data translation. In: Proc. 24th Int. Conf. Very Large Data Bases, VLDB. (1998) 122–133
4. Carroll, J., HP: Matching rdf graphs. In: Proc. in the first International Semantic Web Conference - ISWC 2002. (2002) 5–15
5. Euzenat, J., Valtchev, P.: An integrativive proximity measure for ontology alignment. Proceedings of the workshop on Semantic Integration (2003)
6. Benerecetti, M., Bouquet, P., Zanobini, S.: Soundness of schema matching methods. In Proc. of second European Semantic Web Conference (ESWC 2005). Volume 3532 of LNCS., Heraklion, Crete, Greece, Springer (2005) ISBN 3-540-26124-9.
7. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. Theory, Implementation and Applications. Cambridge University Press (2003)

8. Bouquet, P., Serafini, L., Zanobini, S.: Semantic coordination: a new approach and an application. In Proc. of The Semantic Web – 2nd international semantic web conference (ISWC 2003). Volume 2870 of LNCS., Sanibel Island, Fla., USA (2003)
9. Bouquet, P., Serafini, L., Zanobini, S.: Coordinating semantic peers. In Proced. of AIMSA-2004, Artificial Intelligence: Methodology, Systems, and Applications. Volume 3192 of LNAI., Varna, Bulgaria (2004)
10. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, US (1998)
11. Sceffer, S., Serafini, L., Zanobini, S.: Semantic coordination of hierarchical classifications with attributes. Technical Report 706, DIT, University of Trento (2004) http://eprints.biblio.unitn.it/archive/00000706/.
12. Bunke, H.: Graph matching: Theoretical foundations, algorithms, and applications. In: Proceedings of Vision Interface 2000, Montreal. (2000) 82–88
13. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: The VLDB Journal. (2001) 49–58
14. Bergamaschi, S., Castano, S., Vincini, M.: Semantic integration of semistructured and structured data sources. SIGMOD Record **28**(1) (1999) 54–59
15. Woods, W.: Conceptual indexing: A better way to organize knowledge. Technical Report TR-97-61, Sun Microsystems Laboratories (1997)
16. Kavalec, M., Svatek, V.: Information extraction and ontology learning guided by web directory. In: ECAI Workshop on NLP and ML for ontology engineering, Lyon (2002)
17. Ide, N., Veronis, J.: Introduction to the special issue on word sense disambiguation: the state of the art. Comput. Linguist. **24**(1) (1998) 2–40
18. Agirre, E., Rigau, G.: Word sense disambiguation using conceptual density. In: Proceedings of COLING-96, Copenhagen, Danmark (1996) 16–22
19. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: IJCAI. (1995) 448–453
20. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an algorithm and an implementation of semantic matching. In: Proceedings of ESWS. (2004) 61–75