Peer-to-Peer Semantic Coordination

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

[1] Department of Information and Communication Technology – University of Trento

Via Sommarive, 10 – 38050 Trento (Italy)

[2]ITC-IRST – Istituto per la Ricerca Scientifica e Tecnologica

Via Sommarive, 14 – 38050 Trento (Italy)

bouquet@dit.unitn.it serafini@itc.it zanobini@dit.unitn.it

Abstract

Semantic coordination, namely the problem of finding an agreement on the meaning of heterogeneous schemas, is one of the key issues in the development of the Semantic Web. In this paper, we propose a method for discovering semantic mappings across hierarchical classifications based on a new approach, which shifts the problem of semantic coordination from the problem of computing linguistic or structural similarities (what most other proposed approaches do) to the problem of deducing relations between sets of logical formulae that represent the meaning of concepts belonging to different schema. We show how to apply the approach and the algorithm to an interesting family of schemas, namely hierarchical classifications, and present the results of preliminary tests on two types of hierarchical classifications, web directories and catalogs. Finally, we argue why this is a significant improvement on previous approaches.

1. Introduction

One of the key issues in the development of the Semantic Web is the problem of enabling machines to exchange meaningful information/knowledge across applications which (i) may use autonomously developed schemas (e.g. taxonomies, classifications, database schemas, data types) for organizing locally available data, and (ii) need to discover relations between schemas to achieve their users' goals. This problem can be viewed as a problem of coordination, defined as follows: (i) all parties have an interest in finding an agreement on how to map their schemas onto each others, but (ii) there are many possible/plausible solutions (many alternative mappings across local schemas) among which they need to select the right, or at least a sufficiently good, one. For this reason, we see this as a problem of *semantic coordination*[1].

In environments with more or less well-defined boundaries, like a corporate Intranet, the problem of semantic coordination can be addressed *a priori* by defining and using shared schemas (e.g. ontologies) throughout the entire organization[2]. However, in open environments, like the Semantic Web, this "centralized" approach to semantic coordination is not viable for several reasons, such as the difficulty of "negotiating" a shared model that suits the needs of all parties involved, the practical impossibility of maintaining such a model in a highly dynamic environment, the problem of finding a satisfactory mapping of pre-existing local schemas onto such a global model. In such a scenario, the problem of exchanging meaningful information across locally defined schemas (each possibly presupposing heterogeneous semantic models) seems particularly tough, as we cannot assume an *a priori* agreement, and therefore its solution requires a more dynamic and flexible form of coordination, which we call "peer-to-peer" semantic coordination.

In this paper, we address an important instance of the problem of peer-to-peer semantic coordination, namely the problem of coordinating hierarchical classifications (HCs). HCs are structures having the *explicit* purpose of organizing/classifying some kind of data (such as documents, records in a database, goods, activities, services). The problem of coordinating HCs is significant for at least two main reasons:

- first, HCs are widely used in many applications[3]. Ex-

1 See the introduction of [5] for this notion, and its relation with the notion of *meaning negotiation*.

2 But see [3] for a discussion of the drawbacks of this approach from the standpoint of Knowledge Management applications.

amples are: web directories (see e.g. the Google™ Directory or the Yahoo!™ Directory), content management tools and portals (which often use hierarchical classifications to organize documents and web pages), service registry (web services are typically classified in a hierarchical form, e.g. in UDDI), marketplaces (goods are classified in hierarchical catalogs), PC's file systems (where files are typically classified in hierarchical folder structures);

- second, it is an empirical fact that most actual HCs (as most concrete instances of models available on the Semantic Web) are built using structures whose labels are expressions from the language spoken by the community of their users (including technical words, neologisms, proper names, abbreviations, acronyms, whose meaning is shared in that community). In our opinion, recognizing this fact is crucial to go beyond the use of syntactic (or weakly semantic) techniques, as it gives us the chance of exploiting the complex degree of semantic coordination implicit in the way a community uses the language from which the labels of a HC are taken.

The main technical contribution of the paper is a logic–based algorithm, called CTXMATCH, for coordinating HCs. It takes in input two HCs S and S' and, for each pair of concepts $m \in S$ and $n \in S'$, returns their semantic relation. The relations we consider in this version of CTXMATCH are: m is less general than n, m is more general than n, m is equivalent to n, m is compatible with (possibly overlappings) n, and m is incompatible with (i.e., disjoint from) n. The formal semantics of these relations will be made precise in the paper.

With respect to other approaches to semantic coordination proposed in the literature (often under different "headings", such as schema matching, ontology mapping, semantic integration; see Section 6 for references and a detailed discussion of some of them), our approach is innovative in three main aspects: (1) we introduce a new method for making explicit the meaning of nodes in a HC (and in general, in structured semantic models) by combining three different types of knowledge, each of which has a specific role; (2) the result of applying this method is that we are able to produce a new representation of a HC, in which all relevant knowledge about the nodes (including their meaning in that specific HC) is encoded as a set of logical formulae; (3) mappings across nodes of two HCs are then deduced via logical reasoning, rather then derived through some more or less complex heuristic procedure, and thus can be assigned a clearly defined model-theoretic semantics. As we will show, this leads to a major conceptual shift, as the problem of semantic coordination between HCs is no longer tackled as a problem of computing linguistic or structural similarities (possibly with the help of a thesaurus and of other information about the type of arcs between nodes), but rather as a problem of deducing relations between formulae that represent the meaning of each concept in a given HC. This explains, for example, why our approach performs much better than other ones when two concepts are intuitively equivalent, but occur in structurally very different HCs.

The paper goes as follows. In Section 2 we introduce the main conceptual assumptions of the new approach we propose to semantic coordination. In Section 3 we show how this approach is instantiated to the problem of coordinating HCs. Then we present the main features of CTXMATCH, the proposed algorithm for coordinating HCs (Section 4). In the final part of the paper, we sum-up the results of testing the algorithm on web directories and catalogs (Section 5) and compare our approach with other proposed approaches for matching schemas (Section 6).

2. Our approach

The method we propose assumes that we deal with a network of physically connected entities which can autonomously decide how to organize locally available data; we call these entities *semantic peers*. Peers organize their data using one or more schemas (e.g., database schemas, directories in a file system, classification schemas, taxonomies, and so on); as we said, in this paper we focus on classifications. Different peers may use different schemas to classify the same collection of documents/data, and conversely the same schemas can be used to organize different collections of documents/data.

We also assume that semantic peers need to exchange data (in our scenario, this means documents classified under categories belonging to distinct classification schemas). To do this, each semantic peer needs to discover 'mappings' between its local classification schema(s) and other peers' schemas. Intuitively, a mapping can be viewed as a set of pairwise relations between elements of two distinct classification schemas.

The first idea behind our approach is that *mappings must represent semantic relations*, namely relations with a well-defined model-theoretic interpretation. This is an important difference with respect to approaches based on matching techniques, where a mapping is a measure of (linguistic, structural, ...) similarity between schemas (e.g., a real number between 0 and 1). The main problem with the latter techniques is that the interpretation of their results is an open problem. For example, how should we interpret a 0.9 similarity? Does it mean that one concept is slightly more general than the other one? Or maybe slightly less general?

3 For an interesting discussion of the central role of classification in human cognition see, e.g., [16, 6].

Or that their meaning 90% overlaps (whatever that means)? Instead, our method returns semantic relations, e.g. that the two concepts are (logically) equivalent, or that one is (logically) more/less general, or that they are mutually exclusive. As we will argue, this gives us many advantages, essentially related to the consequences we can infer from the discovery of such a relation[4].

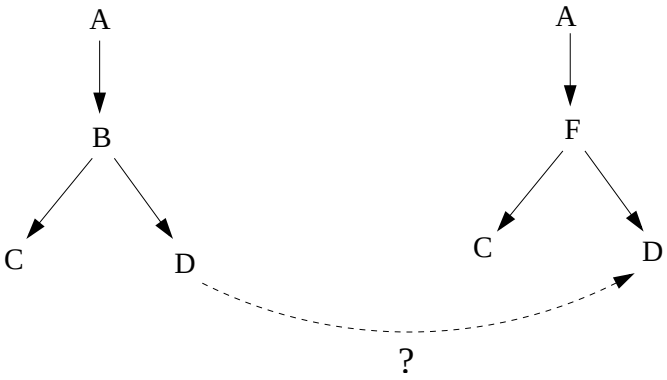

Figure 1. Mapping abstract schemas

The second idea is that, to discover semantic relations, one must make explicit the meaning implicit in each element of a schema. The claim is that making explicit the meaning of elements is the necessary premise for computing semantic relations between elements of distinct schemas, and that this can be done only for schemas in which meaningful labels are used, where "meaningful" means that their interpretation is not arbitrary, but is constrained by the conventions of some community of speakers/users.

To illustrate the consequences of this idea, consider the difference between the problem of mapping abstract schemas (like those in Figure 1) and the problem of mapping schemas with meaningful labels (like those in Figure 2). Nodes in abstract schemas do not have an implicit meaning, and therefore, whatever technique we use to map them, we will find that there is some relation between the two nodes `D` in the two schemas which depends only on the abstract form of the two schemas. The situation is completely different for schemas with meaningful labels. Consider for example the two pairs of structures depicted in Figure 2. Both of them are isomorphic with the pair of abstract schemas depicted in Figure 1. But, despite this similarity, we can easily understand that the relation between the two nodes `MOUNTAIN` is 'less than', while the relation between the two nodes `FLORENCE` is 'equivalent'. Indeed, for the first pair of nodes, the set of documents we would classify under the node `MOUNTAIN` on the left hand side is a subset of the documents we would classify under the node `MOUNTAIN` on the right; whereas the set of documents which we would classify under the node `FLORENCE` in the left schema is exactly the same as the set of documents we would classify under the node `FLORENCE` on the right hand side. The reason of this difference resides in the presence of meaningful labels which allow us to make explicit a lot of information that we have about the terms which appear in the graph, and their relations (e.g., that Tuscany is part of Italy, that Florence is in Tuscany, and so on). It's only this information which allows us to understand why the semantic relation between the two nodes `MOUNTAIN` and the two nodes `FLORENCE` is different.

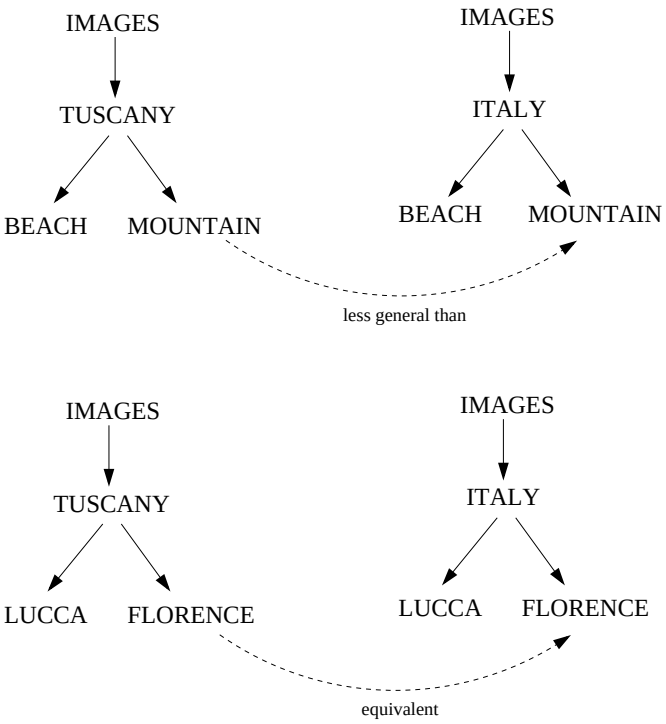

Figure 2. Mapping schemas with meaningful labels

This approach gives us the chance of exploiting the complex degree of semantic coordination implicit in the way a community uses the language from which the labels are taken[5]. The method is based on the explicitation of the

4 For a more detailed discussion of the distinction between syntactic and semantic methods, see [14].

5 Notice that the status of this linguistic coordination at a given time is

meaning associated with each node in a schema (notice that schemas such as the two classifications in Figure 2 *are not* semantic models themselves, as they do not have the purpose of defining the meaning of terms they contain; however, they *presuppose* a semantic model, and indeed that's the only reason why we humans can read them quite easily). The explicitation process uses three different levels of knowledge:

Lexical knowledge: knowledge about the words used in the labels. For example, the fact that the word 'Florence' can be used to indicate 'a city in Italy' or 'a town in northeast South Carolina', and to handle the synonymy;

World knowledge: knowledge about the relation between the concepts expressed by words. For example, the fact that 'Tuscany is part of Italy', or that 'Florence is in Italy';

Structural knowledge: knowledge deriving from how labeled nodes are arranged in a given schema. For example, the fact that the node labeled `MOUNTAIN` is below a node `IMAGES` tells us that it classifies images of mountains, and not, say, books about mountains.

As an example of how the three levels are used, consider again the mapping between the two nodes `MOUNTAIN` of Figure 2. Lexical knowledge is used to determine what concepts can be expressed by each label, e.g. that the word 'Images' can denote the concept 'a visual representation produced on a surface'. World knowledge tells us, among other things, that 'Tuscany is part of Italy'. Finally, structural knowledge tells us that the intended meanings of the two nodes `MOUNTAIN` is 'images of Tuscan mountains' on the left hand side, and 'images of Italian mountains' on the right hand side. Using this information, human reasoners (i) understand the meaning expressed by the left hand node, ('images of Tuscan mountains', denoted by P), (ii) understand the meaning expressed by the right hand node ('images of Italian mountains', denoted by P'), and finally (iii) understand the semantic relation between the meaning of the two nodes, namely that $P \subseteq P'$.

Note that applying the same process for determining the semantic relation holding between the two nodes `FLORENCE`, which belongs to structurally equivalent structures, leads to different result. Indeed, exploiting domain knowledge, we can add the fact that 'Florence is in Tuscany' (such a relation doesn't hold between mountains and Italy in the first example). This further piece of domain knowledge allows us to conclude that, beyond structural similarity, the relation is different (namely, 'equivalent to').

This analysis of meaning has an important consequence on our approach to semantic coordination. Indeed, unlike all other approaches we know of, we do not use lexical knowledge (and, in our case, domain knowledge) to improve the results of structural matching (e.g., by adding synonyms for labels, or expanding acronyms). Instead, we combine knowledge from all three levels to build a new representation of the problem, where the meaning of each node is encoded as a logical formula, and relevant domain knowledge and structural relations between nodes are added to nodes as sets of axioms that capture background knowledge about them.

This, in turn, introduces the last feature of our approach. Indeed, once the meaning of each node, together with all relevant domain and structural knowledge, is encoded as a set of logical formulae, the problem of discovering the semantic relation between two nodes can be stated not as a matching problem, but as a relatively simple problem of logical deduction. Intuitively, as we will say in a more technical form in Section 4, determining whether there is an equivalence relation between the meaning of two nodes becomes a problem of testing whether the first implies the second and vice versa (given a suitable collection of axioms, which acts as a sort of background theory); and determining whether one is less general than the other one amounts to testing if the first implies the second. As we will say, in the current version of the algorithm we encode this reasoning problem as a problem of logical satisfiability, and then compute mappings by feeding the problem to a standard SAT solver.

3. P2P coordination of hierarchical classifications

In this section we show how to apply the general approach described in the previous section to the problem of coordinating HCs. Intuitively, a classification is a grouping of things into classes or categories. When categories are arranged into a hierarchical structure, we have a hierarchical classification. Formally, the hierarchical structures we use to build HCs are *concept hierarchies*, defined as follows in [7]:

Definition 1 (Concept hierarchy) *A* concept hierarchy *is a triple $S = \langle N, E, l \rangle$ where N is a finite set of nodes, E is a set of arcs on N, such that $\langle N, E \rangle$ is a rooted tree, and l is a function from N to a set L of labels.*

Essentially, a concept hierarchy is a rooted tree where categories are nodes of the tree[6]. Given a concept hierar-

already 'codified' in artifacts (e.g., dictionaries, but today also ontologies and other formalized models), which provide senses for words (the set of concepts a word can express) and more complex expressions, relations between concepts, and other important knowledge about them. Our aim is to exploit these artifacts as an essential source of constraints on possible/acceptable mappings across structures.

chy S, a classification can be defined as follows:

Definition 2 (Hierarchical Classification) *A* hierarchical classification *of a set of objects D in a concept hierarchy $S = \langle N, E, l \rangle$ is a function $\mu : N \rightarrow 2^D$.*

We assume that the classification function μ in Definition 2 satisfies the following *specificity principle*: an object $d \in D$ is classified under a node n, if d is about n (according to some criteria, e.g., the semantic intuition of the creator of the classification!) and there isn't a more specific node m under which d could be classified[7].

Prototypical examples of HCs are the web directories of many search engines, as for example the Google™ Directory, the Yahoo!™ Directory, or the Looksmart™ web directory, or the PC's file–systems. A tiny fraction of the HCs corresponding to the Google™ Directory™ and to the Yahoo!™Directory is depicted in Figure 3.

Intuitively, the problem of semantic coordination arises when one needs to find relations between nodes belonging to distinct (and thus typically heterogeneous) HCs. Imagine the following scenario. You have the left lower classification of Figure 2 and you are interested in finding new documents to be classified under the node `FLORENCE`. So you would like to ask the system to find out for you whether there are nodes in different classifications (e.g., the right lower classification of Figure 2) which have the same meaning as, or a meaning related to, the node `FLORENCE` in your classification[8]. Formally, we define the problem of semantic coordination as the problem of discovering mappings between nodes in two distinct classifications S and S':

Definition 3 (Mapping) *A* mapping *M from $S = \langle N, E, l \rangle$ to $S' = \langle N', E', l' \rangle$ is a function $M : N \times N' \rightarrow rel$, where rel is a set of names of accepted semantic relations.*

The set *rel* of accepted semantic relations depends on the intended use of the structures we want to map. Indeed, in our experience, the intended use of a structure (e.g., classifying objects) is semantically much more relevant than the type of abstract structures involved (e.g., tree, graph, ...) to determine how a structure should be interpreted. As the purpose of mapping HCs is to discover relations between nodes (concepts) that are used to classify objects, four possible relations can hold between two nodes m and n belonging to different HCs: $m \xrightarrow{\supseteq} n$ (m is more general than n); $m \xrightarrow{\subseteq} n$ (m is less general than n); $m \xrightarrow{\equiv} n$ (m is equivalent to n); $m \xrightarrow{\perp} n$ (m is disjoint from n). Note that the relations are essentially the set theoretical relations, with the exception of the intersection.

6 Hereafter node, category and classes are used in the same sense.

7 See for example Yahoo!™ instruction for "Finding an appropriate Category" at http://docs.yahoo.com/info/suggest/appropriate.html.

8 Similar examples apply to catalogs. Here we use web directories, as they are well-known to most readers and easy to understand.

The formal semantics of these expressions is given in terms of compatibility between documents occurring in two different concept hierarchies S_s and S_t[9]:

Definition 4 *A mapping function M from $S_s = \langle N_s, E_s, l_s \rangle$ to $S_t = \langle N_t, E_t, l_t \rangle$ is* extensionally correct *with respect to two hierarchical classifications μ_s and μ_t of the same set of documents D in S_s and S_t, respectively, if the following conditions hold for any node $n_s \in N_s$ and $n_t \in N_t$:*

$$
\begin{array}{lcl}
n_s \xrightarrow{\supseteq} n_t & \Rightarrow & \mu_s(n_s\downarrow) \supseteq \mu_t(n_t\downarrow) \\
n_s \xrightarrow{\subseteq} n_t & \Rightarrow & \mu_s(n_s\downarrow) \subseteq \mu_t(n_t\downarrow) \\
n_s \xrightarrow{\perp} n_t & \Rightarrow & \mu_s(n_s\downarrow) \cap \mu_t(n_t\downarrow) = \emptyset \\
n_s \xrightarrow{\equiv} n_t & \Rightarrow & \mu_s(n_s\downarrow) = \mu_t(n_t\downarrow)
\end{array}
$$

where $\mu(c\downarrow)$ is the union of $\mu(d)$ for any d in the subtree rooted at c.

If no relation can be computed, the algorithm returns what we call a compatibility relation ($m \xrightarrow{*} n$, m is compatible with n), which is interpreted as *possible intersection*: it means that there is at least one interpretation under which the concepts associated with the two nodes (and therefore the classified documents) overlap[10].

4. The algorithm: CTXMATCH

The CTXMATCH algorithm (see algorithm 1) takes as input two HCs (representing the structural knowledge), a lexicon L (representing the lexical knowledge) and an ontology O (representing world knowledge), and returns as output a mapping between the nodes of the two classifications.

9 The semantics provided in Definition 4 can be viewed as a special case of *Local Models Semantic*, namely the semantic provided in [12, 4] to capture the (compatibility) relations between local representations called *contexts*. Proof-theoretically, compatibility relations are formalized as *bridge rules*, namely rules whose premise and conclusions belong to different theories [13]. CTXMATCH can be viewed as a first attempt of automatically discovering bridge rules (compatibility relations) across contexts.

10 The reason why we introduced the relation called *possible intersection* is that we wanted to have a placeholder for potential relations that we can't deduce – but we can't reject either – based on the available knowledge. For example, 'Images of dogs' and 'Images of cats' will be evaluated as compatible unless we have positive information that dogs and cats are disjoint. Of course, one might say that this conclusion is wrong, but we need to take into account that in other cases (e.g., 'Images of churches' and 'Images of monuments') it would be right (for the sake of argument, let us imagine that there is no explicit connection between churches and monuments in the world knowledge in use). So the decision is between reasoning under a "closed world assumption" (which would prevent us from finding any relation in both cases), or reasoning under an "open world assumption" (which would find a relation of compatibility in both cases). Of course, the open world assumption increases the recall, but decreases the precision of the algorithm, whereas the closed world assumption would have the opposite effect. Our decision was to go for an open world solution, as most concrete ontologies are not rich enough to support the conjecture that anything which cannot be deduced is false.

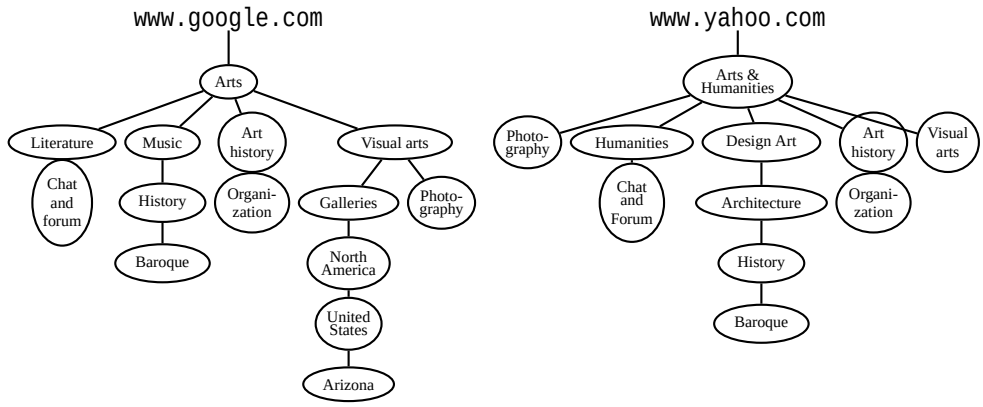

Figure 3. Examples of concept hierarchies (source: Google!Directory and Yahoo!Directory)

For the sake of simplicity, in the explanation of the algorithm, we imagine that the two HCs taken as input are the two structures depicted in the lower hand of Figure 2.

The algorithm has essentially the following two main macro steps.

Semantic Explicitation (Steps 2–3): in this phase, the meaning of each node m in a classification S is made explicit using information from the lexicon L and the ontology O. Explicitation is performed by generating a formula ϕ approximating the meaning expressed by a node in a HC, and a set of axioms Θ formalizing the relevant world knowledge. Consider, for example, the node FLORENCE in lower left HC hand of Figure 2: steps 2–3 will generate a formula ϕ approximating the statement 'Images of Florence in Tuscany' and a set of axioms Θ approximating the statement 'Florence is in Tuscany'. The pair $\langle \phi, \Theta \rangle$, called *contextualized concept*, expresses the meaning of a labeled node in a structure with respect to L and O.

Semantic comparison (Step 4) : In this phase the problem of finding the semantic relation between two nodes m and n is encoded as the problem of finding the semantic relation holding between two contextualized concepts, $\langle \phi, \Theta \rangle$ and $\langle \psi, \Upsilon \rangle$, associated with the nodes m and n respectively in the semantic explicitation phase. To prove that the two nodes FLORENCE in Figure 2 are equivalent, we deduce the logical equivalence between the formulas associated with the nodes, approximating respectively the statements 'Images of Florence in Tuscany' (ϕ) and 'Images of Florence in Italy' (ψ), given some domain axioms, as formulas approximating the statements 'Florence is a city of Tuscany' (Θ), 'Florence is a city of Italy' (Υ), and 'Tuscany is a region of Italy'.

Algorithm 1 CTXMATCH(S,$S'$$L, O$)

▷ *Hierarchical Classifi cations* S, S'
▷ *Lexicon* L
▷ *Ontology* O
VarDeclaration:
contextualized concept $\langle \phi, \Theta \rangle, \langle \psi, \Upsilon \rangle$
semantic relation r
mapping M

1 **for** each pair of nodes $m \in S$ and $n \in S'$ **do**
2 $\langle \phi, \Theta \rangle \leftarrow$ SEMANTIC−EXPLICITATION(m, S, L, O);
3 $\langle \psi, \Upsilon \rangle \leftarrow$ SEMANTIC−EXPLICITATION(n, S', L, O);
4 $r \leftarrow$ SEMANTIC−COMPARISON$(\langle \phi, \Theta \rangle, \langle \psi, \Upsilon \rangle, O)$;
5 $M \leftarrow M \cup \langle m, n, r \rangle$;
6 **return** M;

Finally, step 5 generates the mapping simply by reiteration of the same process over all pairs of nodes $m \in S$ and $n \in S'$ (step 1) and step 6 returns the mapping.

The two following sections describe in detail these two top-level functions SEMANTIC−EXPLICITATION and SEMANTIC−COMPARISON.

In the version of the algorithm presented here, we use WORDNET[11] as a source of both lexical and domain knowledge. However, WORDNET could be replaced by another combination of a linguistic and a world knowledge resources.

4.1. Semantic explicitation

The function SEMANTIC−EXPLICITATION (algorithm 2) has the main goal of making explicit in a formula the mean-

11 WORDNET [11] is a well-known Lexical/Ontological repository which contains the set of concepts possibly denoted by a word (called synsets, i.e. set of synonyms), and a set of relations (essentially *Is-A* and *Part-Of*) holding between senses. Hereafter, the notion of *sense* is used as synonym of *concept*.

ing of a labeled node into a structure, by means of lexical and world knowledge. The formula will be expressed in same logical language. The choice of the logical language depends on how expressive one wants to be in the approximation of the meaning of nodes, and on the complexity of the NLP techniques used to process labels. In our first implementation, we adopted propositional logic, where each propositional letter corresponds to a concept (synset) provided by WORDNET.

Algorithm 2 SEMANTIC−EXPLICITATION(t,S,L,O)
▷ *t is a node in S*
▷ *structure S*
▷ *lexicon L*
▷ *world knowledge O*

VarDeclaration:
focus F
set of concepts $con[]$
set of axioms Σ
set of simple concepts Γ
formula δ

1 $F \leftarrow$ focus(t,S)
2 **for** each node n in F **do**
3 $con[n] \leftarrow$ EXTRACT−CANDIDATE−CONCEPTS(n,L);
4 $\Sigma \leftarrow$ EXTRACT−LOCAL-AXIOMS$(F,con[],O)$;
5 $con[] \leftarrow$ FILTER−CONCEPTS$(F,\Sigma,con[])$;
6 **for** each node n in the path from t to root in F **do**
7 $\Gamma \leftarrow \Gamma \cup$ BUILD−SIMPLE−CONCEPT$(con[],n)$;
8 $\delta \leftarrow$ BUILD−COMPLEX−CONCEPT(Γ);
9 **return** $\langle \delta, \Sigma \rangle$;

Determining the meaning of a node n in a structure is a process that doesn't require to consider all the other nodes. So, we need to concentrate only to the set of nodes *relevant* for building the meaning of n.

To this end, we introduce the notion of *focus* of a node n in a classification S, denoted by $f(n,S)$. Intuitively, the focus is the smallest sub-tree of S that one should take into account to determine the meaning of n in S. In CTXMATCH, the focus is defined as follows:

Definition 5 (Focus) *The* focus *of a node n in a classification $S = \langle N,E,l \rangle$, is a finite concept hierarchy $f(n,S) = \langle N',E',l' \rangle$ such that: $N' \subseteq N$, and N' contains exactly n and its children, its ancestors, and all their children; $E' \subseteq E$ is the set of edges between the concepts of N'; l' is the restriction of l on N'.*

This definition of focus is motivated by observations on how humans read HCs. When searching for documents in a HC, we incrementally construct the meaning of a node n by navigating the classification from the root to n. During this navigation, we have access to the labels of the ancestors of n, and also to the labels of their siblings. This information is used at each stage to build the meaning expressed by a node in a structure[12].

Step 1 of function SEMANTIC−EXPLICITATION determines the focus of a node in a structure.

Steps 2 and 3 exploit lexical knowledge to associate with each word occurring in the nodes of the focus all the possible concepts denoted by the word itself. When two or more words in a label are contained in the Lexicon as a single expression (a so-called multiword), the corresponding sense is selected. Consider the lower left structure of Figure 2. The label 'Florence' is associated with two concepts (or senses), provided by the lexicon (in this case, WORDNET), corresponding to 'a city in central Italy on the Arno' (`florence#1`) or a 'a town in northeast South Carolina' (`florence#2`). In order to maximize the possibility of finding an entry into the Lexicon, we use both a postagger and a lemmatizator over the labels.

In the step 4, the function EXTRACT−LOCAL−AXIOMS exploits ontology with the aim of finding (possible) ontological relations existing between concepts associated with the words of the labels in a focus. Consider again the left lower structure of Figure 2. Imagine that the concept 'a region in central Italy' (`tuscany#1`) has been associated with the node `TUSCANY` in step 3. The function EXTRACT−LOCAL−AXIOMS checks if there exists some relation between the concept `tuscany#1` and the concepts associated with other nodes, namely `florence#1` and `florence#2` (associated with node `FLORENCE`), `images#1`, ..., `images#8` (associated with node `IMAGE`), and `lucca#1` (associated with node `LUCCA`)[13]. For example, exploiting world knowledge, we can discover, among other things, that '`florence#1` PartOf `tuscany#1`', i.e. that there exists a 'part of' relation between the first sense of 'Florence' and the first sense of 'Tuscany'.

World knowledge relations derived from WORDNET are translated into logical axioms, according to Table 1. So, the relation '`florence#1` PartOf `tuscany#1`' is encoded as '`florence#1` $\rightarrow$ `tuscany#1`'.

To sum up, step 4 tries to discover the relationships holding between concepts expressed by labels of nodes in the focus $f(n,S)$ of some node n.

Step 5 has the goal of filtering out unlikely senses associated with each node. Going back to the previous example, we try to discard one of the senses associated with the node `FLORENCE`. Intuitively, the sense 2 of 'Florence', 'a town in northeast South Carolina' (`florence#2`), can be tenta-

12 This definition of focus is appropriate for HCs. With structures used for different purposes, different definitions of focus should be used. For example, if a concept hierarchy is used to represents data–type, the meaning of a node is determined also by the meaning of its sub-nodes, so a more suitable definition of focus $f(n,H)$ would include for example the sub-tree rooted at n.

13 In this example the focus of the node `FLORENCE` corresponds to the entire structure.

WORDNET relation	axiom
s#k synonym t#h	s#k $\equiv$ t#h
s#k { hyponym | PartOf }t#h	s#k $\rightarrow$ t#h
s#k { hypernym | HasPart }t#h	t#h $\rightarrow$ s#k
s#k antonyms t#h	$\neg$(t#k $\wedge$ s#h)

Table 1. WORDNET relations and their axioms.

tively discarded, because the node FLORENCE refers clearly to the city in Tuscany. We reach this result by analyzing the extracted local axioms: the presence of an axiom such as 'florence#1 $\rightarrow$ tuscany#1' is used to make the conjecture that the contextually relevant sense of Florence is the city in Tuscany, and not the city in USA. When ambiguity persists (axioms related to different senses or no axioms at all), all the possible senses are left.

Steps 6–7 have the main goal of providing an interpretation of the meaning expressed by (the labels of) the nodes *independently from the position they occur into the structure*. Let F be a focus for a node n in a concept hierarchy H. In this phase we associate with each node t in the path from root to n a logical formula, that we call *simple concept*, representing all possible linguistic interpretations of the label of t allowed by the lexical knowledge available.

Labels are processed by text chunking (via Alembic chunker [9]), and connectives are translated into a logical form according to the following rules:

- coordinating conjunctions and commas are interpreted as disjunctions;
- prepositions, like 'in' or 'of', are interpreted as a conjunction;
- expressions denoting exclusion, like 'except' or 'but not', are interpreted as negations.

In the example we provide in Figure 2, the simple concept approximating the meaning expressed by the node IMAGES is the formula image#1 $\vee \ldots \vee$ image#8 (the eight senses provided by WORDNET), the meaning expressed by the node TUSCANY is the atom tuscany#1 (the only sense provided by WORDNET), while the meaning of the node FLORENCE is approximated by the atom florence#1 (one of the two senses provided by WORDNET and not discarded by the filtering).

More complicated cases can be handled. Consider the two classifications depicted in Figure 3. The following *simple concepts* are found:

Google

- the simple concept expressed by the node Baroque is baroque#1, the unique sense of the word 'Baroque' presents in WORDNET;
- the simple concept expressed by the node Arizona is arizona#1, one of the two senses of 'Arizona' (arizona#2 in the sense of 'glossy snake' has been discarded because of the presence of the node UNITED STATES);
- the simple concept expressed by the node Chat and Forum is chat#1 $\vee$ chat#2 $\vee$ chat#3 $\vee$ forum#1 $\vee$ forum#2 $\vee$ forum#3, i.e. the disjunction of the meaning of 'chat' and 'forum' taken separately (both 'chat' and 'forum' have tree senses in WORDNET);
- the simple concept expressed by the node North America is north america#1 $\vee$ north america#2, the senses associated with the multiword 'North America' in Lexicon (WORDNET).

Yahoo

- the simple concept expressed by node the Visual Arts is visual art#1 $\wedge$ $\neg$photography#1: both Visual Arts and Photography are sibling nodes under Arts & Humanities; since in WORDNET the concept photography#1 is in a *IsA* relationship with the concept visual art#1, the node Visual arts is re-interpreted as visual arts with the exception of photography.

Step 8 has the goal of building the formula approximating the meaning expressed by a node *into a structure*, that we call *complex concept*. In this version of the algorithm, we chose to build the complex concept expressed by a node n as the conjunction of the simple concepts associated with all of its ancestors (i.e., the path from root to n). So, the complex concept associated with the node FLORENCE is (image#1 $\vee \ldots \vee$ image#8) $\wedge$ tuscany#1 $\wedge$ florence#1.

Again, this choice depends on the intended use of the structure. From the *specificity principle* of classifications (see section 3) follows that the documents classified under a node n can be, conceptually, classified also under the ancestors of n (in absence of n). For example, considering the Figure 2, the images classified under the path IMAGES/TUSCANY/FLORENCE could be classified, in absence of the node FLORENCE, also under the node IMAGES/TUSCANY. This means that the meaning associated with a node n, say FLORENCE, that we call ϕ, must be a subset of the meaning associated with an ancestor node m, say TUSCANY, that we call ψ. So, formally, $\phi \subseteq \psi$. Rephrasing this in propositional logics, it means that $\phi \rightarrow \psi$. A possible way of formalizing this aspect in propositional logics is to use conjunctions, so that, for example, if the complex concept associated with the node TUSCANY is (image#1 $\vee \ldots \vee$ image#8) $\wedge$ tuscany#1 (say, ψ), then the complex concept associated with the node FLORENCE must

be $(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{tuscany\#1} \wedge \texttt{florence\#1}$ (say, ϕ). In this case we have that $\phi \rightarrow \psi$.

Finally, step 9 returns the *contextualized concept*, namely the formula expressing the meaning of the node and the set of local axioms returned by step 4.

4.2. Semantic comparison

After semantic explicitation is performed, the problem of discovering semantic relations between two nodes *m* and *n* in two HCs can be reduced to the problem of checking if a logical relation holds between two formulas ϕ and ψ: this is checked as a problem of propositional satisfiability (SAT), and then computed via a standard SAT solver.

Algorithm 3 SEMANTIC–COMPARISON($\langle\phi,\Theta\rangle$,$\langle\psi,\Upsilon\rangle$,$O$)

▷ *contextualized concept* $\langle\phi,\Theta\rangle$
▷ *contextualized concept* $\langle\psi,\Upsilon\rangle$
▷ *world knowledge O*

VarDeclaration:
set of formulas Γ
semantic relation *r*

1 $\Gamma \leftarrow$ EXTRACT–RELATIONAL–AXIOMS(ϕ, ψ, O);
2 **if** $\Theta,\Upsilon,\Gamma \models \neg(\phi \wedge \psi)$ **then** $r \leftarrow$ '$\perp$';
3 **else if** $\Theta,\Upsilon,\Gamma \models (\phi \equiv \psi)$ **then** $r \leftarrow$ '$\equiv$';
4 **else if** $\Theta,\Upsilon,\Gamma \models (\phi \rightarrow \psi)$ **then** $r \leftarrow$ '$\subseteq$';
5 **else if** $\Theta,\Upsilon,\Gamma \models (\psi \rightarrow \phi)$ **then** $r \leftarrow$ '$\supseteq$';
6 **else** $r \leftarrow$ '$*$';
7 **return** *r*;

In Step 1, the function EXTRACT–RELATIONAL–AXIOMS tries to find axioms which connect concepts belonging to different HCs. The process is the same as that of the function EXTRACT–LOCAL–AXIOMS, described above, with the only difference that it involves concepts (senses, if we use WORDNET) belonging to different HCs. Consider, for example, the senses `italy#1` and `tuscany#1` associated respectively with nodes `ITALY` and `TUSCANY` of Figure 2: the relational axioms express the fact that, for example, 'Tuscany is part of Italy', and it is translated in the axiom $\texttt{tuscany\#1} \rightarrow \texttt{italy\#1}$, according to Table 1.

The problem of finding the semantic relation between two nodes *n* and *m* (step 2) is encoded into a satisfiability problem involving the contextualized concepts associated with two nodes of different structures and the relational axioms extracted in the previous phases. As we said before, five possible semantic relations are allowed: disjoint ($\perp$), equivalent ($\equiv$), less than ($\subseteq$), more than ($\supseteq$) and compatible ($*$). Steps 2–6 check which relation holds by running the SAT solver on five satisfiability problems, where the sets of formulas Θ,Υ,Γ represent the local axioms of the source node, the local axioms of the target node and the relational axioms respectively, and ϕ and ψ represent the *complex concepts* approximating the meaning expressed by the source node and the target node respectively. Note that the compatibility relation is leaved as default case (step 6).

Going back to our example, to prove whether the two nodes labeled `FLORENCE` in Figure 2 are equivalent, we check the logical equivalence between the formulas approximating the meaning of the two nodes, given the local and the relational axioms. Formally, we have the following satisfiability problem:

Θ	$\texttt{florence\#1} \rightarrow \texttt{tuscany\#1}$
ϕ	$(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{tuscany\#1} \wedge \texttt{florence\#1}$
Υ	$\texttt{florence\#1} \rightarrow \texttt{italy\#1}$
ψ	$(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{italy\#1} \wedge \texttt{florence\#1}$
Γ	$\texttt{tuscany\#1} \rightarrow \texttt{italy\#1}$

It is easy to see that the returned relation is '$\equiv$'. Note that the satisfiability problem for finding the semantic relation between the nodes `MOUNTAIN` of Figure 2 is the following:

Θ	$\emptyset$
ϕ	$(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{tuscany\#1} \wedge \texttt{mountain\#1}$
Υ	$\emptyset$
ψ	$(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{italy\#1} \wedge \texttt{mountain\#1}$
Γ	$\texttt{tuscany\#1} \rightarrow \texttt{italy\#1}$

The returned relation is '$\subseteq$'.

The algorithm has been run not only in *ad hoc* examples, as the classifications depicted in Figure 2, but also in real world examples. In the following table we present some results obtained in running CTXMATCH for finding relations between the nodes of the portion of Google and Yahoo classifications depicted in Figure 3.

Google node	Yahoo node	Relation found
`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 ($\equiv$)

In the first example, CTXMATCH 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, CTXMATCH 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' ($\texttt{photography\#1} \rightarrow \texttt{visual art\#1}$). 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 photography is one of its siblings). Given this information, it easy to deduce that, although despite the two nodes lye 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 $\rightarrow$ 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. Testing the algorithm

In this section, we report from [18] some results of the first test on CTXMATCH on real HCs (i.e., pre-existing classifications used in real applications).

5.1. Use case Product Re-classification

In order to centrally manage all the company acquisition processes, the headquarter of a well known worldwide telecommunication company had realized an e-procurement system[14], which all the company branch-quarters were required to join. Each single office was also required to migrate from the product catalogue they used to manage, to this new one managed within the platform. This catalogue is extracted from the Universal Standard Products and Services Classification (UNSPSC), which is an open global coding system that classifies products and services. UNSPSC is used extensively around the world in electronic catalogues, search engines, procurement application systems and accounting systems. UNSPSC is a four-level hierarchical classification; an extract is reported in the following table:

Level 1 **Furniture and Furnishings**
Level 2 **Accommodation furniture**
Level 3 **Furniture**
Level 4 **Stands**
Level 4 **Sofas**
Level 4 **Coat racks**

The Italian office asked us to apply the matching algorithm to re-classify into UNSPSC (version 5.0.2) the catalogue of office equipment and accessories they used to classify company suppliers. The result of running CTXMATCH over UNSPSC and the catalogue can be clearly interpreted in terms of re-classification: if the algorithm returns that the item *i* of the catalogue is equivalent to, or more specific than, the node c_{UNSPSC} of UNSPSC, then *i* can be classified under c_{UNSPSC} of UNSPSC.

The items to be re-classified are mainly labeled with Italian phrases, but labels also contain abbreviations, acronyms, proper names, some English phrases and some typing errors. The English translation of an extract of this list is reported in the following table. The italic text were contained in the original labels.

Code	Description
ENT.21.13	cartridge *hp desk jet 2000c*
ENR.00.20	magnetic tape cassette *exatape 160m xl 7,0gb*
ESA.11.52	*hybrid roller pentel* red
EVM.00.40	safety scissors, length 25 cm

The item list was matched against two UNSPSC's segments: *Office Equipment and Accessories and Supplies* (segment 44) and *Paper Materials and Products* (segment 14).

Notice that the company item catalogue we had to deal with was a plain list of items, each identified with a numerical code composed of two numbers, the first referring to a set of more general categories. For example, the number 21 at the beginning of *21.13-cartridge hp desk jet 2000c* corresponds to *printer tapes, cartridge and toner*. We first normalized and matched the plain list against UNSPSC. This did not lead us to a satisfactory result. The algorithm performed much better when we made explicit the hierarchical classification implicitly contained in the item codes. This was done by substituting the first numerical code of each item with their textual description provided by experts of the company.

After running CTXMATCH, the validation phase of our results was made by comparing them with the results of a simple keyword-based algorithm. Obviously, in order to establish the correctness of results in terms of precision and recall we must compare them with a correct and complete matching list. Not having such a list, we asked a domain expert to manually validate them.

5.2. Results

This section presents the results of the re-classification phase. Consider first the baseline matching process. The baseline was performed by a simple keyword-based matching that worked according to the following rule:

> *for each item description (made up of one or more words) return the set of nodes, and their paths, which maximize the occurrences of the item words*

The following tables summarizes the results of the baseline matching:

14 An e-procurement system is a technological platform which supports a company in managing its procurement processes and, more in general, the re-organization of the value chain on the supply side.

ITEMS	UNSPSC CLASSES	RELATION
drill/drill with 2-4 holes	Office machines, materials and accessories/ Supplies for office / Supplies for writing-desks/ Paper staplers	$\subseteq$
printing tape, toner, cartridge , printing head	Office machines, materials and accessories/ Supplies of printing, telecoping-machine e coping-machine/ Ink cartridges	$\subseteq$
printing tape, toner, cartridge , printing head	Office machines, materials and accessories Printing forniture, telecoping-machine e coping-machine/ Toner	$\subseteq$
lampostil pen, marker, highlighter/highlighter	Supplies for office/ Tools for writing/ Highlighters	$\subseteq$
ball-point pen, pen	Supplies for office/ Tools for writing/ Assortment of pens and penciles	$*$
double ruler/double ruler in white plastic	Accessories for the office and the writing desk/ Accessories for drawing	$*$
cleaning kit/PC cleaning kit	Office machines, materials and accessories/ Accessories for office machines/ computer cleaning kit	$*$
cleaning kit / /cleaning kit for exatape 4 mm tape heads	Materials for office / Office machines, materials and accessories/ Accessories for office machines/ Cleaners of tapes	$*$

Table 2. Some examples of matching found by CTXMATCH and not found by the baseline.

	Baseline classification	
Total items	194	100%
Rightly classified	75	39%
Wrongly classified	92	47%
Non classified	27	14%

Given the 193 items to be re-classified, the baseline process found 1945 possible nodes in UNSPSC. This means that for each item the baseline found an average of 6 possible reclassifications. What is crucial is that only 75 out of the 1945 proposed nodes are correct. The percentage of error is quite high (47%) with respect to the one of correctness (39%). The results of the matching algorithm are reported in the following table:

	CTXMATCH **classification**	
Total items	194	100%
Rightly classified	136	70%
Wrongly classified	16	8%
Non classified	42	22%

In this case, the percentage of success is higher (70%) and, even more relevant, the percentage of error is minimal (8%)[15]. This is also confirmed by the values of precision and recall, computed with respect to the validated list:

	Total matches	**Precision**	**Recall**
Baseline	1945	4%	39%
CTXMATCH	641	21%	70%

The baseline precision level is quite small, while the matching one is not excellent, but definitely better. The same observations can be made also for the recall values.

Table 2 reports some examples where the algorithm found a correct item for re-classification, while the baseline did not.

If there are not enough information to infer semantic relation, CTXMATCH returns a percentage, which is intended to represent the degree of compatibility between the two elements. Degree of compatibility is computed on the basis of a linguistic co-occurrence measures. Examples of compatibility relations are contained in the last four rows of Table 2.

As far as the Non Classified items, notice that:

- In some cases, the item to be re-classified were not correctly classified in the company catalogue. Therefore, CTXMATCH could not compute the relations with the node and its parent node, in the right way. Examples are: *ashtray* was classified under *tape dispenser*; *wrapping paper* was classified under *adhesive labels*.
- In other cases, semantic coordination was not discovered due to a lack of domain knowledge. For instance

15 Notice that the algorithm did not take into account only the UNSPSC level 4 category, since in some cases catalogues items can be matched with UNSPSC level 3 category nodes.

to match *paper for hp* with UNSPSC class of printer paper it would have been necessary to know that *hp* stands for Helwett Packard, and that it is a company which produces printers.

In a further experiment, we run CTXMATCH between the company catalogue (in Italian) and the English version of UNSPSC. This was possible because the matching is computed on the basis of the WORDNET sense IDs, and in the version of WORDNET we used, wordnet-senses ID of Italian and English words are aligned (i.e., the wordnet-sense ID associated with word and its translation in the other language is the same). This experiment allows us to find more semantic matches.

More in general, using aligned lexical sources allows us to approach and manage multi-language environments and to exploit the richness which typically characterizes the English version of linguistic resources[16].

6. Related work

CTXMATCH shifts the problem of semantic coordination from the problem of matching (in a more or less sophisticated way) semantic structures (e.g., schemas) to the problem of deducing semantic relations between sets of logical formulae. Under this respect, to the best of our knowledge, there are no other works to which we can compare ours.

However, it is important to see how CTXMATCH compares with the performance of techniques based on different approaches to semantic coordination. There are four other families of approaches that we will consider: graph matching, automatic schema matching, semi-automatic schema matching, and instance based matching. For each of them, we will discuss the proposal that, in our opinion, is more significant. The comparison is based on the following five dimensions: (1) if and how structural knowledge is used; (2) if and how lexical knowledge is used; (3) if and how domain knowledge is used; (4) if instances are considered; (5) the type of result returned. The general results of our comparison are reported in Table 3.

In graph matching techniques, a concept hierarchy is viewed as a tree of labelled nodes, but the semantic information associated with labels is substantially ignored. In this approach, matching two graphs G_1 and G_2 means finding a sub-graph of G_2 which is isomorphic to G_2 and report as a result the mapping of nodes of G_1 into the nodes of G_2. These approaches consider only structural knowledge and completely ignore lexical and domain knowledge. Some examples of this approach are described in [22, 21, 20, 19, 15].

16 The results of this experiment is not reported as they are not comparable with our simple keyword-based baseline, which makes no sense with multiple languages.

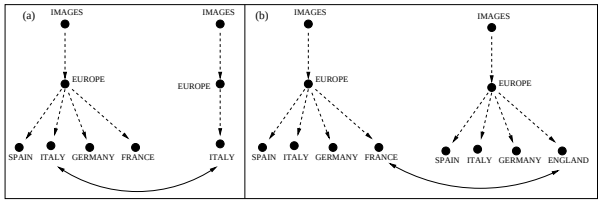

Figure 4. Example of a correct mapping not found by CUPID and of a wrong mapping found by CUPID

CUPID [17] is a completely automatic algorithm for schema matching. Lexical knowledge is exploited for discovering linguistic similarity between labels (e.g., using synonyms), while the schema structure is used as a matching constraint. That is, the more the structure of the subtree of a node s is similar to the structure of a subtree of a node t, the more s is similar to t. For this reason CUPID is more effective in matching concept hierarchies that represent data types rather than hierarchical classifications. With hierarchical classifications, there are cases of equivalent concepts occurring in completely different structures, and completely independent concepts that belong to isomorphic structures. Two simple examples are depicted in Figure 4. In case (a), CUPID does not match the two nodes labelled with `ITALY`; in case (b) CUPID finds a match between the node labelled with `FRANCE` and `ENGLAND`. The reason is that CUPID combines in an additive way lexical and structural information, so when structural similarity is very strong (for example, all neighbor nodes do match), then a relation between nodes is inferred without considering labels. So, for example, `FRANCE` and `ENGLAND` match because the structural similarity of the neighbor nodes is so strong that labels are ignored.

MOMIS (Mediator envirOnment for Multiple Information Sources) [1] is a set of tools for information integration of (semi-)structured data sources, whose main objective is to define a global schema that allow an uniform and transparent access to the data stored in a set of semantically heterogeneous sources. One of the key steps of MOMIS is the discovery of overlappings (relations) between the different source schemas. This is done by exploiting knowledge in a Common Thesaurus together with a combination of clustering techniques and Description Logics. The approach is very similar to CUPID and presents the same drawbacks in matching hierarchical classifications. Furthermore, MOMIS includes an interactive process as a step of the integration procedure, and thus, unlike CTXMATCH, it does not support a fully automatic and run-time generation of mappings.

GLUE [10] is a taxonomy matcher that builds mappings taking advantage of information contained in instances, us-

	graph matching	CUPID / MOMIS	GLUE	CTXMATCH
Structural knowledge	•	•		•
Lexical knowledge		•	•	•
Domain knowledge			•	•
Instance-based knowledge			•	
Type of result	Pairs of nodes	Similarity measure $\in [0..1]$ between pairs of nodes	Similarity measure $\in [0..1]$ between pairs of nodes	Semantic relations between pairs of nodes

Table 3. Comparing CTXMATCH with other methods

ing machine learning techniques and domain-dependent constraints, manually provided by domain experts. GLUE represents an approach complementary to CTXMATCH. GLUE is more effective when a large amount of data is available, while CTXMATCH performs better when less data are available, or the application requires a quick, on-the-fly mapping between structures. So, for instance, in case of product classification such as UNSPSC or Eclss (which are pure hierarchies of concepts with no data attached), GLUE cannot be applied. Combining the two approaches is a challenging research topic, which can probably lead to a more precise and effective methodology for semantic coordination.

7. Conclusions and future work

In this paper we presented a new approach to semantic coordination in open and distributed environments, and an algorithm (called CTXMATCH) that implements this method for hierarchical classifications. The algorithm has already been used in a peer-to-peer application for distributed knowledge management (the application is described in [2]), and is going to be applied in a peer-to-peer wireless system for ambient intelligence [8].

An important lesson we learned from this work is that methods for semantic coordinations should not be grouped together on the basis of the type of abstract structure they aim at coordinating (e.g., graphs, concept hierarchies), but on the basis of the intended use of the structures under consideration. In this paper, we addressed the problem of coordinating concept hierarchies when used to build hierarchical classifications. Other possible uses of structures are: conceptualizing some domain (ontologies), describing services (automata), describing data types (schemas). This "pragmatic" level (i.e., the use) is essential to provide the correct interpretation of a structure, and thus to discover the correct mappings with other structures.

The importance we assign to the fact that HCs are labelled with meaningful expressions does not mean that we see the problem of semantic coordination as a problem of natural language processing (NLP). On the contrary, the solution we provided is mostly based on knowledge representation and automated reasoning techniques. However, the problem of semantic coordination is a fertile field for collaboration between researchers in knowledge representation and in NLP. Indeed, if in describing the general approach one can assume that some linguistic meaning analysis for labels is available and ready to use, we must be very clear about the fact that real applications (like the one we described in Section 4) require a massive use of techniques and tools from NLP, as a good automatic analysis of labels from a linguistic point of view is a necessary precondition for applying the algorithm to HC in local applications, and for the quality of mappings resulting from the application of the algorithm.

The work we presented in this paper is only the first step of a very ambitious scientific project, namely to investigate the minimal common ground needed to enable communication between autonomous entities (e.g., agents) that cannot "look into each others head", and thus can achieve some degree of semantic coordination only through other means, like exchanging messages or examples, pointing to things, remembering past interactions, generalizing from past communications, and so on. In this context, we are currently working on the second version of the algorithm, which will include the following new features: (i) the extension of its application beyond hierarchical classifications (namely to structures whose purposes is not to classify things, e.g. service descriptions); (ii) the generalization of the types of structures that can be coordinated (e.g., graphs, data type descriptions, ...); (iii) the usage of lexical and world knowledge sources different from WORDNET (our short term goal is to wrap any OWL-based ontology and link it to WORDNET as a lexical source); (iv) the introduction of a more expressive logic (e.g., Descrip-

tion Logic) to formalize the concepts associated with nodes (as an example, considering the node MOUNTAIN of the upper right hand structure of Figure 2, we want the SEMANTIC–EXPLICITATION function to extract the following DL concept: '$\texttt{image} \sqcap \exists \texttt{about}.(\texttt{mountain} \sqcap \exists \texttt{located}.\texttt{italy})$', namely 'images concerning mountains located in Italy'); consequently, we plan to move from boolean SAT solvers to DL reasoners; (v) the possibility of using different lexical and/or world knowledge sources for each of the local structures to be coordinated. The last problem is probably the most challenging one, as it introduces asymmetric mappings (if different peers use different lexical and world knowledge, the mappings they compute between their classifications may be different). In our opinion, such a scenario opens a completely new area of research, in which semantic agreements cannot be viewed simply as the result of a coordination process (computation of mappings), but will be *negotiated* among parties with potentially conflicting interests.

References

[1] Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.

[2] M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori. Kex: a peer-to-peer solution for distributed knowledge management. In D. Karagiannis and U. Reimer, editors, *Fourth International Conference on Practical Aspects of Knowledge Management (PAKM-2002)*, Vienna (Austria), 2002.

[3] M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. managerial and technological implications. *Novatica and Informatik/Informatique*, III(1), 2002.

[4] A. Borgida and L. Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In R. Meersman and Z. Tari, editors, *On The Move to Meaningful Internet Systems 2002: CoopIS, Doa, and ODBase*, volume 2519 of *LNCS*, pages 36–53. Springer Verlag, 2002.

[5] P. Bouquet, editor. *AAAI-02 Workshop on Meaning Negotiation*, Edmonton, Canada, July 2002. AAAI, AAAI Press.

[6] G. C. Bowker and S. L. Star. *Sorting things out: classification and its consequences*. MIT Press., 1999.

[7] A. Büchner, M. Ranta, J. Hughes, and M. Mäntylä. Semantic information mediation among multiple product ontologies. In *Proc. 4th World Conference on Integrated Design & Process Technology*, 1999.

[8] P. Busetta, P. Bouquet, G. Adami, M. Bonifacio, and F. Palmieri. K-Trek: An approach to context awareness in large environments. Technical report, Istituto per la Ricerca Scientifica e Tecnologica (ITC-IRST), Trento (Italy), April 2003. Submitted to UbiComp'2003.

[9] D. S. Day and M. B. Vilain. Phrase parsing with rule sequence processors: an application to the shared CoNLL task. In *Proc. of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September 2000.

[10] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proceedings of WWW-2002, 11th International WWW Conference, Hawaii*, 2002.

[11] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, US, 1998.

[12] C. Ghidini and F. Giunchiglia. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence*, 127(2):221–259, April 2001.

[13] F. Giunchiglia and L. Serafini. Multilanguage Hierarchical Logics or: how we can do without modal logics. *Artificial Intelligence*, 65(1):29–70, 1994.

[14] F. Giunchiglia and P. Shvaiko. Semantic matching. *Proceedings of the ISWC-03 workshop on Semantic Integration*, Sanibel Island (Florida), October 2003.

[15] Jeremy Carroll Hewlett-Packard. Matching rdf graphs. In *Proc. in the first International Semantic Web Conference - ISWC 2002*, pages 5–15, 2002.

[16] G. Lakoff. *Women, Fire, and Dangerous Things*. Chicago Unversity Press, 1987.

[17] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58, 2001.

[18] B. M. Magnini, L. Serafini, A. Doná, L. Gatti, C. Girardi, , and M. Speranza. Large–scale evaluation of context matching. Technical Report 0301–07, ITC–IRST, Trento, Italy, 2003.

[19] Tova Milo and Sagit Zohar. Using schema matching to simplify heterogeneous data translation. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 122–133, 24–27 1998.

[20] Marcello Pelillo, Kaleem Siddiqi, and Steven W. Zucker. Matching hierarchical structures using association graphs. *Lecture Notes in Computer Science*, 1407:3–??, 1998.

[21] Jason Tsong-Li Wang, Kaizhong Zhang, Karpjoo Jeong, and Dennis Shasha. A system for approximate tree matching. *Knowledge and Data Engineering*, 6(4):559–571, 1994.

[22] K. Zhang, J. T. L. Wang, and D. Shasha. On the editing distance between undirected acyclic graphs and related problems. In Z. Galil and E. Ukkonen, editors, *Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching*, volume 937, pages 395–407, Espoo, Finland, 1995. Springer-Verlag, Berlin.