Automatic Ontology Matching Via Upper Ontologies: A Systematic Evaluation

Viviana Mascardi, Angela Locoro, Paolo Rosso

Abstract—"Ontology matching" is the process of finding correspondences between entities belonging to different ontologies. This paper describes a set of algorithms that exploit upper ontologies as semantic bridges in the ontology matching process and presents a systematic analysis of the relationships among features of matched ontologies (number of simple and composite concepts, stems, concepts at the top level, common English suffixes and prefixes, ontology depth), matching algorithms, used upper ontologies, and experiment results. This analysis allowed us to state under which circumstances the exploitation of upper ontologies gives significant advantages with respect to traditional approaches that do no use them. We run experiments with SUMO-OWL (a restricted version of SUMO), OpenCyc and DOLCE. The experiments demonstrate that when our "structural matching method via upper ontology" uses an upper ontology large enough (OpenCyc, SUMO-OWL), the recall is significantly improved while preserving the precision obtained without upper ontologies. Instead, our "non structural matching method" via OpenCyc and SUMO-OWL improves the precision and maintains the recall. The "mixed method" that combines the results of structural alignment without using upper ontologies and structural alignment via upper ontologies improves the recall and maintains the F-measure independently of the used upper ontology.

Index Terms—Ontology Matching, Upper Ontology.

1 Introduction

In a paper that dates back to 2001, J. A. Hendler predicted that "*in the next few years virtually every company, university, government agency or ad hoc interest group will want their web resources linked to ontological content - because of the many powerful tools that will be available for using it*" [20]. Hendler's vision has found a partial realisation: ontologies, web services, and the combination of both, i.e., semantic web services, are increasingly exploited to share knowledge within and outside the boundaries of companies and other organisations.

However, although probably most companies, universities, and government agencies *would want* to link their web resources to a common ontological content, few of them *have* already implemented this vision.

The delay with respect to Hendler's predictions has a theoretical explanation in the impossibility theorem applied to ontologies: when independent and autonomous organisations need to select one common ontology in order to interoperate, "*no ontology can be maximum for all individuals and the group, i.e. some individuals or the group will lose when an ontology is adopted over some other ontology"* [31].

Suggestions for overcoming the impediments in the use of one single shared ontology in an integrated knowledge-based system come from consolidated research results by the database community. There, the addressed issue was to support semantic heterogeneity in a federation of autonomous, heterogeneous, loosely-coupled database systems, allowing each of them to keep its own database and its own independence. In that situation, the impossibility theorem does not apply: the problem is not to agree upon one single database that meets the needs of all the federated systems, but to provide each federated system with the means to interoperate with others. One solution ([19], [23], [37]) is to find the *correspondences between objects that model similar information* in distinguished database systems and allow each federated system to exploit these correspondences for exchanging information. To fully support interoperability, the federated database systems must share a *set of commonly understood concepts and relationships among them* which may be structured in different ways: a "semantic dictionary" [19], a "dynamic classificational ontology" [23], a "federated schema" [37]. This common knowledge is used as a "semantic bridge" between the systems to be integrated, which keep using their own underlying databases.

The set of commonly understood concepts may be designed before the integration takes place, and may be developed by hand. This is the solution adopted by [19] and [37] where a "sharing advisor" and the user are in charge of defining the semantic dictionary and the federated schema, respectively. The dynamic classificational ontology [23] consists of one portion built by a human administrator, and another automatically derived from it.

In order to keep the human out of the loop, an alternative approach is to draw the set of commonly understood concepts and relationships from some existing repository containing general concepts that are the same across all domains. In the ontology field, ontologies of this kind are known as "upper ontologies" [47]. Within a group of interoperating organisa-

• *Viviana Mascardi is with DISI, Università degli Studi di Genova, Via Dodecaneso 35, 16146, Genova, Italy, E-mail:* `viviana.mascardi@unige.it`
Angela Locoro is with DIBE, Università degli Studi di Genova, Via All'Opera Pia 11a, 16145 Genova, Italy, E-mail: `angela.locoro@unige.it`
Paolo Rosso is with Natural Language Engineering Lab., DSIC, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022, Valencia Spain, E-mail: `prosso@dsic.upv.es`

tions, the upper ontology would only be exploited for boosting the retrieval of semantically meaningful correspondences. No organisation would be asked to substitute its own domain ontology with it, thus agreeing on it should not raise the problems discussed in [31].

This paper is about the usage of existing upper ontologies for improving the results of *automatic* ontology matching, namely the activity of automatically finding correspondences between entities belonging to two or more ontologies.

Many events and publications address ontology matching. Starting from 2004, an Ontology Alignment Contest (now named Ontology Alignment Evaluation Initiative, OAEI, http://oaei.ontologymatching.org/) is run every year with the purpose of establishing a consensus for evaluating schema matching and ontology integration methods. The Ontology Matching portal, http://www.ontologymatching.org, lists more than 250 publications and provides free access to all of them, and in 2007 the book *Ontology Matching* [10] was published. The use of ontologies in computer science dates back to the early nineties [17]. The problem of finding correspondences among them was raised about seven years later, although similar problems had been addressed many years before in the database community as discussed above. Semi-automatic ontology matching and merging systems appeared since 1998 ([22], [30], [28], just to cite some of the oldest ones) but the first formalisation of the ontology matching problem dates back to 2000 [5]. As an effect of such a growing interest in the ontology matching problem, many matching techniques have been proposed including those that use background knowledge and domain ontologies for improving the matching results. Surprisingly, very few attempts to exploit "upper ontologies" in a systematic way have been made so far. The work described in this paper consists of a systematic and repeatable experimentation on the use of a selected set of upper ontologies as bridges for improving the results of the matching process, together with a careful analysis of the obtained results.

The paper is organised in the following way: Section 2 introduces the ontology matching problem, upper ontologies, and existing proposals to use background knowledge for boosting the ontology matching process. Section 3 discusses the algorithms we used, whereas Section 4 describes the methodology we followed for running our experiments, the ontologies we matched, and the results we obtained. Section 5 concludes and outlines the future directions of our work.

2 Related Work

To the best of our knowledge, no concrete attempts to exploit upper ontologies to face the ontology matching problem have been reported in the literature, apart from our preliminary work described in [26] and LOM, a Lexicon-based Ontology Mapping Tool [25].

LOM uses four methods to match the vocabularies from any two ontologies: (1) whole term matching; (2) word constituent matching; (3) synset matching; and (4) type matching.

Type matching exploits the mappings from WordNet synsets to the SUMO and MILO ontologies (see Section 2.2), for the source terms that are unmatched in the first three methods.

The main difference between our approach and LOM's one, is that ours can be adopted with any upper ontology which is given as an input parameter to our matching algorithm, whereas LOM is based on SUMO. Although we are not aware of other approaches like LOM's one and ours for ontology matching via upper ontologies, research on both ontology matching and upper ontologies is extremely lively and produced important results. Having some knowledge on these results is needed to understand our approach.

This section discusses the state-of-the-art of ontology matching techniques and tools, and describes the upper ontologies used in our experiments: Cyc, DOLCE and SUMO. Finally, this section analyses existing proposals of exploiting background knowledge for boosting ontology matching.

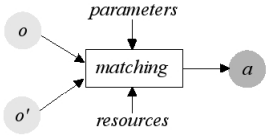
2.1 Ontology Matching

In the sequel we give an account of the concepts that we will use throughout the paper and of the metrics that we used for computing our alignments. We follow the terminology proposed in [10] and we adopt the same definitions given there, as well as the same symbols within figures, simplifying them for our purposes if it is the case.

Definition 2.1. (Matching Process). *A matching process can be seen as a function f which takes two ontologies o and o', a set of parameters p and a set of oracles and resources r, and returns an alignment A between o and o'.*

A schematic representation of this process is given in Figure 1.

Figure 1. Matching process, adapted from [11]



Definition 2.2. (Correspondence). *A correspondence between an entity e belonging to ontology o and an entity e' belonging to ontology o' is a 5-tuple $< id, e, e', R, conf >$ where:*

- *id is a unique identifier of the correspondence;*
- *e and e' are the entities (e.g. properties, classes, individuals) of o and o' respectively;*
- *R is a relation such as "equivalence", "more general", "disjointness", "overlapping", holding between the entities e and e'.*
- *$conf$ is a confidence measure (typically in the $[0, 1]$ range) holding for the correspondence between the entities e and e';*

In our experiments, we only considered classes as entities and equivalence as relation.

Definition 2.3. (Alignment). *An alignment of ontologies o and o' is a set of correspondences between entities of o and o'.*

Very often, the main activity of a matching method is to measure a pair-wise similarity between entities and to compute the best match between them. These methods exploit the definitions of similarity and of distance, and may be roughly classified into "name-based", "structure-based", "extensional" and "semantic-based" according to the kind of input they operate on. We only use name-based techniques that enable to measure how much an entity of an ontology is related to an entity of another ontology by comparing the names of the entities themselves. In particular, we adopt both string-based and language-based methods.

Definition 2.4. (Similarity measure). *A similarity measure $\sigma : o \times o \rightarrow \mathbb{R}$ is a function from a pair of entities to a real number expressing the similarity between two objects such that:*

$$\forall x, y \in o, \sigma(x, y) \geq 0 \quad \textit{(positiveness)}$$
$$\forall x, y, z \in o, \sigma(x, x) \geq \sigma(y, z) \quad \textit{(maximality)}$$
$$\forall x, y \in o, \sigma(x, y) = \sigma(y, x) \quad \textit{(symmetry)}$$

The dissimilarity is the dual operation of the similarity.

Definition 2.5. (Distance). *A distance (or metric) $\delta : o \times o \rightarrow \mathbb{R}$ is a dissimilarity function such that:*

$$\forall x, y \in o, \delta(x, y) = 0 \textit{ iff } x = y \quad \textit{(definiteness)}$$
$$\forall x, y, z \in o, \delta(x, y) + \delta(y, z) \geq \delta(x, z) \quad \textit{(triangular ineq.)}$$

A (dis)similarity is normalised if it ranges over the interval $[0, 1]$ of real numbers.

The methods introduced below use normalised measures[1].

String-based methods. These methods measure the similarity of two entities just looking at the strings (seen as mere sequences of characters) that label them. They include:

Substring Distance: measures the ratio of the longest common substring of two strings with respect to their length.

n-gram Distance: two strings are the more similar, the more n-grams (sequences of n characters) in common they have [7].

SMOA Measure: similarity of two strings is a function of their commonalities (in terms of substrings) as well as of their differences [39].

We used the above measures for implementing our algorithms. The experiments reported in [39] demonstrate that they are the most stable[2] ones if compared to others such as *Levenstein*, *Jaro-Winkler*, *Monge-Elkan*, *Smith-Waterman*, and *Needleman-Wunsch*.

Language-based methods. These methods exploit natural language processing techniques to find the similarity between two strings seen as meaningful pieces of text rather than sequences of characters. Some of these methods exploit external resources, such as WordNet [27], that provide semantic relations such as synonymy, hyponymy, hypernymy, to compute the similarity.

1. In its original formulation, SMOA returns a value in $[-1, 1]$, but most existing implementations map this value to the $[0, 1]$ interval.

2. "Stability" is defined as the ability of a string metric to perform almost optimal even if small divergences from the optimal threshold take place.

Many implemented tools and algorithms for ontology matching exist. Chapter 6 of [10] analyses 50 systems divided into schema-based (25), instance-based (12), mixed (11), and meta-matching (2). Discussing them is out of the scope of this paper.

2.2 Upper Ontologies

There are few implemented upper ontologies: BFO [16], Cyc [24], DOLCE [11], GFO [21], PROTON [8], Sowa's ontology [38], and SUMO [29].

Synoptic tables for comparing them are provided in Appendix A. The choice of Cyc, DOLCE and SUMO for carrying our experiments out is motivated by their dimension: together with PROTON, they are the largest upper ontologies available. As our experiments demonstrated, we expected that using a small upper ontology would not help the matching process and thus dimension was the first criterion we followed in our choice. However, while PROTON is no longer maintained, Cyc, DOLCE and SUMO are extremely lively projects.

Cyc: The Cyc Knowledge Base (KB), http://www.cyc.com/, developed by Cycorp, is a formalised representation of facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. The KB consists of terms and assertions which relate those terms. These assertions include both simple ground assertions and rules. The Cyc KB is divided into thousands of "microtheories" focused on a particular domain of knowledge, a particular level of detail, a particular interval in time, etc. Cyc is a commercial product, but Cycorp also released OpenCyc (http://www.cyc.org/), the open source version of the Cyc technology, and ResearchCyc (http://research.cyc.com/), namely the Cyc ontology delivered with a research-only license. Both OpenCyc and ResearchCyc are more limited than the commercial version. The Commercial Cyc KB (including Cyc's microtheories) contains more than 300,000 concepts and nearly 3,000,000 assertions (facts and rules), using more than 15,000 relations. Instead, OpenCyc includes about 26,000 concepts, 4,800 data properties, and 62,000 individuals. However, unlike proprietary Cyc, all of the rules which define these terms are omitted in OpenCyc. Cyc is represented in the CycL formal language (http://www.cyc.com/cycdoc/ref/cycl-syntax.html). The latest release of Cyc includes an Ontology Exporter that allows to export specified portions of Cyc to OWL [44] files. Cyc has been used in the domains of natural language processing, in particular for the tasks of word sense disambiguation and question answering, of network risk assessment, and of representation of terrorism-related knowledge. The last release of Cyc (as well as of OpenCyc and ResearchCyc) includes links between Cyc concepts and about 12,000 WordNet synsets.

DOLCE: DOLCE (a Descriptive Ontology for Linguistic and Cognitive Engineering), http://www.loa-cnr.it/DOLCE.html, has been developed by the researchers of the Laboratory for Applied Ontology, headed by N. Guarino. DOLCE is the first module of the WonderWeb Foundational Ontologies Library. It has a clear cognitive bias, in the sense that it aims at capturing the ontological categories underlying natural language and human common sense. According to DOLCE,

different entities can be co-located in the same space-time. DOLCE is described by its authors as an "ontology of particulars", which the authors explain as meaning an ontology of instances, rather than an ontology of universals or properties. The taxonomy of the most basic categories of particulars assumed in DOLCE includes, for example, abstract quality, abstract region, amount of matter, physical quality, temporal region. DOLCE is implemented in First Order Logic, KIF [3], and OWL. Its OWL version contains around 250 concepts and 320 properties, divided into 10 sub-ontologies. DOLCE is used in many projects (http://www.loa-cnr.it/dolcevar.html) that include multilingual information retrieval from legal databases, semantic web techniques for improving the retrieval of learning material, semantic-based knowledge flow system for the European home textiles industry. DOLCE-Lite-Plus has been aligned to about 100 WordNet synsets (http://www.loa-cnr.it/ontologies/OWN/OWN.owl). The OWL version of DOLCE can be freely downloaded from http://www.loa-cnr.it/ontologies/DLP_397.owl.

SUMO: SUMO (Suggested Upper Merged Ontology), http://www.ontologyportal.org/, was initially designed at Teknowledge Corporation by I. Niles and A. Pease, with a contribution by C. Menzel. SUMO and its domain ontologies [29] form one of the largest formal public ontology in existence today. They are being used for research and applications in search, linguistics and reasoning. SUMO is extended with many domain ontologies and a complete set of links to WordNet, and is freely available. SUMO consists of SUMO itself (the official latest version on the IEEE web site can be downloaded from www.ontologyportal.org), the MId-Level Ontology (MILO), and ontologies of Communications, Countries and Regions, Distributed Computing, Economy, Finance, Engineering Components, Geography, Government, Military, North American Industrial Classification System, People, Physical Elements, Transnational Issues, Transportation, Viruses, World Airports. Additional ontologies of terrorism are available on request. SUMO contains about 1,000 terms and 4,000 axioms; if we consider also the terms and axioms of its domain ontologies, however, it reaches the dimension of 20,000 terms and 60,000 axioms. SUMO is implemented in the first-order logic language SUO-KIF (http://suo.ieee.org/SUO/KIF/suo-kif.html) that can be automatically translated into OWL, although the translation is lossy. The applications of SUMO are documented by hundreds of published papers describing its use (http://www.ontologyportal.org/Pubs.html). The largest user community is in linguistics, but other classes of applications include "pure" representation and reasoning. Applications range from academic to government, to industrial ones. Since May 2007, SUMO has been mapped to all of WordNet 3.0 by hand. SUMO is free and owned by the IEEE. The ontologies that extend SUMO are available under GNU General Public License.

2.3 Ontology matching with background ontologies

The work closest to ours is the one reported in [2] that discusses the experiments carried out for matching the anatomy-related portions of CRISP (The Computer Retrieval of Information on Scientific Projects, http://crisp.cit.nih.gov/) and MeSH (The Medical Subject Headings (MeSH), http://www.nlm.nih.gov/mesh/) using the FMA ontology (The Foundational Model of Anatomy ontology, http://sig.biostr.washington.edu/projects/fm/) as background knowledge. Aleksovski and his colleagues performed five experiments. First of all, a direct alignment of CRISP and MeSH was computed, based on string-based and structural methods. Then, both CRISP and MeSH were aligned with FMA, using the same string-based and structural methods, and the obtained alignments were used to induce an alignment between CRISP and MeSH. The results obtained by the authors showed that matching via FMA found many more *narrower-than* and *broader-than* relations between couples of concepts in CRISP and MeSH than the direct matching. FMA provided the knowledge needed for finding relationships that, otherwise, could not be found by only analysing CRISP and MeSH.

These experiments are less general than ours, both because the FMA ontology is not an upper ontology, and because they are restricted to a specific domain. Nevertheless, the conclusion they lead to is coherent with our results: by increasing the recall and keeping a comparable precision, *"using comprehensive background knowledge in form of ontology can boost the ontology matching process as compared to a direct matching of the two ontologies."* Similar experiments, leading to the same results, are described in [1] and [40]. None of these three works exploits upper ontologies, and all of them apply to a specific application domain, the medical one.

The use of textual and lexical resources, WordNet in particular, as background knowledge has been proposed by many researchers, [41], [15], [34], [6], [13], [14]. An original proposal comes from [35] who discuss the exploitation of the semantic resources available online. In [36] a Conceptual Ontological Graph is used to model concepts representing the key terms extracted from a document and to perform sophisticated text retrieval tasks based on term semantics rather than on the classical term frequency. Since we concentrate on ontology matching via background ontologies (upper ontologies in our case), approaches based on textual, lexical, and semantic web resources fall out of the scope of our paper.

3 Algorithms for Automatic Ontology Matching via Upper Ontologies

For demonstrating the advantages of using upper ontologies in the matching process, we implemented the uo_match, $structural\_uo\_match$, and $mixed\_match$ algorithms. $Structural\_uo\_match$ looks at the identity between $c \in o$ and a super-concept of $c' \in o'$ (and vice versa), or between super-concepts of c and c', to decide whether c and c' are related; uo_match does not look at the ontology structure. Both of them exploit upper ontologies as bridges between o and o'. $Mixed\_match$ aggregates the results obtained by $structural\_uo\_match$ and $structural\_parallel\_match$, a matching algorithm discussed in Section 3.1 that does not use upper ontologies.

The algorithms we implemented compute correspondences between concepts only. We created reference alignments that

only match concepts, and we discarded correspondences between individuals and between properties from the alignments computed by our algorithms. The reason of our choice is that finding correspondences between properties in a meaningful way requires to also take their domain and range into account. For example, the property "has_pet" between a kid and a domesticated animal has not the same meaning as the property "has_pet" between a patient and the results of a Positron Emission Tomography scanning, and they should not be related in any way. In a similar way, computing a correspondence between individuals would require to consider their class into account. It makes no sense to match the "apple" individual in o, and the "apple" individual in o', if they are instances of "fruit" and "computerBrands", respectively. The API we used for computing similarity measures cannot take domain, range, and class into account in the correct way. This also explains our choice of discarding property restrictions when exploring the sub- and super-classes, and considering only subClassOf relations with simple entities used as their property values. Implementing algorithms that overcome these limitations is recognised to be an hard task, and is not our principal objective.

When computing an alignment, different ontology mismatches may arise. In two papers that date back to 1997 [42] and 1998 [43], P. R. S. Visser et al. classify them into (1) *conceptualisation mismatches* that arise when two (or more) conceptualisations of a domain differ in the ontological concepts distinguished or in the way these concepts are related and (2) *explication mismatches* due to differences in the way the conceptualisation is specified (abstracting from the ontology implementation language). The process of solving ontology mismatches is also named "ontology reconciliation" and it "*is generally a human-mediated process, [...] because most of the decisions on how to resolve the kinds of ontological mismatches [...] require a human to identify that different symbols represent the same concept, or that the same symbols represent different concepts*" [18].

Our algorithms are fully automatic and hence they cannot cope with all the possible ontology mismatches. In particular, they do not face conceptualisation mismatches, but we are not aware of any automatic tool that copes with them in a satisfactory way. Instead, our algorithms can solve many (even if not all) *Term and Definiens* and *Term* mismatches. These mismatches arise when two concepts are synonyms. The exploitation of WordNet allows our algorithms to create correspondences between c and c' if both of them belong to WordNet and they are defined as synonyms there. This happened for example in the third test we run (see Section 4.2), where the correspondence between *Sea* in the Geofile ontology and *Ocean* in the Space ontology was found because they are synonyms in WordNet. The usage of upper ontologies may correctly match two concepts that have the same meaning but that were not recognised as synonyms in WordNet. Referring to the third test again, the $structural\_uo\_match$ algorithm used with SUMO-OWL allowed us to find the correspondence between *School* in Geofile and *Educational_structure* in Space which was discovered neither by the WordNet-based method, nor by string-based ones. On the other hand, our algorithms cannot recognise *Concept and Definiens* and *Concept* mismatches that arise when two concepts are homonyms. In fact, concepts with the same syntax are always matched with confidence 1. In the ten tests we run, however, concepts represented by the same string always had the same meaning (for example, *Book* in Ka and *Book* in Bibtex, in the first test).

This section describes our matching algorithms and their implementation. In section 3.1 we describe the auxiliary functions we developed for implementing them. Uo_match is discussed in Section 3.2, $structural\_uo\_match$ in Section 3.3, and $mixed\_match$ in Section 3.4. Section 3.5 provides some details on their implementation.

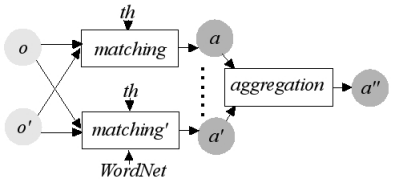
3.1 Auxiliary functions

Uo_match is based on three functions: $aggregate(a, a')$, $parallel\_match(o, o', res, par)$, and $compose(a, a')$, where o and o' are ontologies, a and a' are alignments, res are external resources, and par are parameters. All the three functions return an alignment.

$Aggregate(a, a')$ produces the alignment obtained by making the union of all the correspondences in a and a', and choosing the correspondence with highest confidence measure, in case the same correspondence[3] belongs to both a and a'.

$Parallel\_match(o, o', res, par)$ computes an alignment between o and o' by applying substring, n-gram, SMOA, and language-based methods introduced in Section 2.1 in parallel, as suggested in [10, Chap. 5.1], and aggregating them. The only external resource we use is WordNet ($res = \{WordNet\}$), which is given in input to the language-based method, and the only parameter is a configurable threshold in $[0, 1]$ used for discarding correspondences that are not relevant, since their confidence is lower than it ($par = \{th\}$). In order to simulate the parallelism of the aggregation process, $aggregate$ is initially called on the first two alignments obtained by running the first two matching algorithms; the output is then aggregated with the third alignment, and so on. Figure 2 gives a representation of this process. We may notice that not all the matching methods take WordNet as input. In fact, substring, n-gram, SMOA do not use external resources.

Figure 2. Parallel matching: $parallel\_match(o, o', \{WordNet\}, \{th\})$.



$Compose(a, a')$ computes the alignment a'' in such a way that a correspondence $\langle id, c, c', r, conf\rangle$ belongs to a'' iff $\exists\ c_u$ such that $\langle id_1, c, c_u, r, conf_1\rangle \in a, \langle id_2, c', c_u, r, conf_2\rangle \in a'$, and $conf = conf_1 * conf_2$. In our algorithm the inputs of $compose$, a and a', are alignments between o and the upper

3. "Same" apart from the correspondence identifier.

ontology u, and o' and u, respectively. Thus, the second concept c_u in the correspondences belongs to u. If both $c \in o$ and $c' \in o'$ correspond to the same concept $c_u \in u$, then c and c' are related via r with confidence $conf_1 * conf_2$. The choice of $conf_1 * conf_2$ as confidence of the resulting alignment ensures that the confidence remains in $[0, 1]$, and that initial high confidences lead to a resulting confidence which is still high, whereas at least one low confidence leads to a low resulting confidence.

Besides the $parallel\_match$ function, we implemented the $structural\_parallel\ \_match$ function that implements a sort of "structure-based extension" of the alignment a output by $parallel\_match$. $Structural\_parallel\_match$ adds to a those correspondences $\langle id, c, c', r, conf\rangle$ such that

- either c is identical to one of c''s super-concepts (or vice versa, c' is identical to one of the super-concepts of c),
- or c and c' have two super-concepts, say $s \in o$ and $s' \in o'$ respectively, identical.

In a similar way, we implemented a $structural\_compose$ function that composes two alignments considering correspondences involving super-concepts into account. We prefixed the names of these functions with $structural$ because they look at the structure of the matched ontologies.

$Structural\_compose$ takes a decay factor $df \in [0, 1]$, used to measure the "confidence decay" as we consider relations that involve super-concepts. In our experiments, df has been set to 0.5. It also takes the upper ontology u used as the reference ontology for computing a and a' as input, since it must be navigated in order to find the super-concepts of a given concept.

$Structural\_compose(a, a', u, df)$ computes the alignment a'' between o and o' via u in such a way that a correspondence $\langle id, c, c', r, conf\rangle$ belongs to a'' if either

- $\exists\ c_u$ such that $\langle id_1, c, c_u, r, conf_1\rangle \in a, \langle id_2, c', c_u, r, conf_2\rangle \in a'$, and $conf = conf_1 * conf_2$, or
- $\exists\ c_u, c'_u$ such that $\langle id_1, c, c_u, r, conf_1\rangle \in a, \langle id_2, c', c'_u, r, conf_2\rangle \in a'$, c'_u is a super-concept of c_u in u, and $conf = conf_1 * conf_2 * df$ (note the confidence decay multiplicative factor), and vice versa.
- $\exists\ c_u, c'_u$ such that $\langle id_1, c, c_u, r, conf_1\rangle \in a, \langle id_2, c', c'_u, r, conf_2\rangle \in a'$, c_u and c'_u have a common super-concept in u, and $conf = conf_1 * conf_2 * df^2$ (note the power factor applied to the confidence decay).

3.2 Uo_match algorithm

The $uo\_match(o, o', u, res, par)$ function just calls

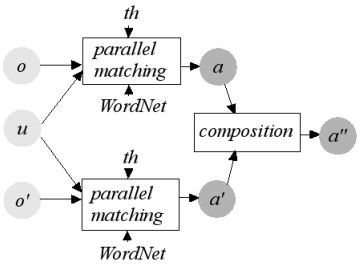
$compose$
$(parallel\_match(o, u, res, par),$
$parallel\_match(o', u, res, par))$

with $res = \{WordNet\}$ and $par = \{th\}$ (Figure 3). The th parameter has been set to 0.5 in our experiments, and u is an upper ontology.

3.3 Structural_uo_match algorithm

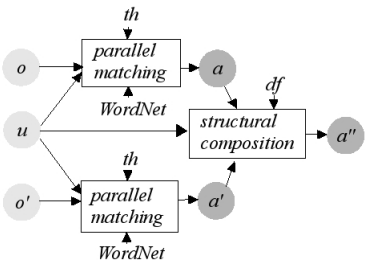
Figure 4 represents the $structural\_uo\_match$ function, defined as

Figure 3. Matching via upper ontologies: $uo\_match(o, o', u, \{WordNet\}, \{th\})$.



$structural\_compose($
$parallel\_match(o, u, WordNet, th),$
$parallel\_match(o', u, WordNet, th),$
$\{u\}, \{df\}).$

Figure 4. Structural matching via upper ontologies: $structural\_uo\_match(o, o', u, \{WordNet\}, \{df, th\})$.



The $structural\_uo\_match$ uses the $parallel\_match$ function for computing the alignments, and not the $structural\_parallel\_match$ one. The exploitation of the structure is deferred to the composition stage. The reason why we also defined a $structural\_parallel\_match$ function is that we want to compare homogeneous matching methods. Thus, in our experiments, we compared the results of uo_match with those of $parallel\_match$ and the results of $structural\_uo\_match$ with those of $structural\_parallel\_match$.

3.4 Mixed_match algorithm

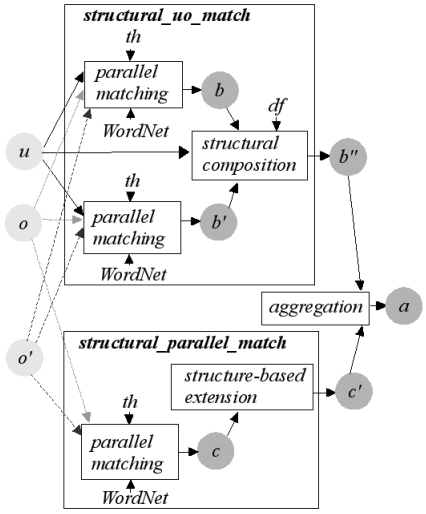
A $mixed\_match$ algorithm obtained by aggregating the alignment output by the $structural\_parallel\_match$ algorithm (direct matching exploiting structure), and the one output by the $structural\_uo\_match$ algorithm (matching via upper ontology, exploiting structure), has also been implemented.

Figure 5 graphically depicts all the activities to be performed in order to obtain the alignment $a = mixed\_match(o, o', u, \{WordNet\}, \{df, th\})$.

3.5 Implementation details

We extended the implementation of the Alignment API version 3.1, delivered on February, the 5th, 2008, and available form http://alignapi.gforge.inria.fr/ under GNU Lesser General Public License, with our matching methods. The methods offered

Figure 5. Mixed matching:
$mixed\_match(o, o', u, \{WordNet\}, \{df, th\})$.



by the Alignment API, as well as our new methods, accept ontologies expressed in OWL, RDF [46], RDFS [45].

Among the methods offered by the API, we used **StringDistAlignment**, that provides the `subStringDistance`, `ngramDistance`, and `smoaDistance` string metric methods, and **JWNLAlignment**, that computes a substring distance between the entity names of the first ontology and the entity names of the second ontology expanded with WordNet 3.0 synsets. To compare the performance of WordNet 3.0 w.r.t. WordNet 2.0, we run experiments using both versions. Table 6 shows the differences in the results we obtained.

We implemented a **SubSupClassAlignment** method that finds correspondences between $c \in o$ and $c' \in o'$ by looking at string equality between one of them, and one super-concept of the other one, or between super-concepts of both. We also implemented the methods for alignment aggregation and composition.

In order to discard property restrictions when exploring the sub- and super-classes, we pre-processed the ontologies given in input to our algorithms and removed property restrictions from them. We used the JENA parser, http://jena.sourceforge.net/, for the pre-processing activities.

4 Experimentation

This section discusses the experiments carried out for understanding if, and under which circumstances, using upper ontologies as bridges for matching o and o' may prove useful. Our experiments demonstrate that:

- With respect to the results obtained by non structural direct matching, the usage of non structural methods with SUMO-OWL and OpenCyc leads to an improvement of the average precision (+9.7% and +16% respectively), whereas the recall and F-measure experience a small degradation (-1.2% with SUMO-OWL, between -1.8% and -1.7% with OpenCyc).
- With respect to the results obtained by structural direct matching, the usage of structural methods with SUMO-OWL and OpenCyc leads to an improvement of the average recall (+2.8% with SUMO-OWL and +2.9% with OpenCyc). Precision degrades of -1.7/-1.8% and F-measure increases of 0.3/0.5%.
- The usage of structural and non structural methods with DOLCE degrades precision, recall, and F-measure with respect to structural and non structural direct matching.
- Combining direct matching and matching via DOLCE, SUMO-OWL and OpenCyc always boosts recall: in all the tests we run, the best recall was obtained by the mixed matching method (with respect to direct matching, the average improvement was +6.6% when using SUMO-OWL, +7.1% with OpenCyc and +2.5% with DOLCE). This good result is paid with a degradation of the precision (-7.7% with SUMO-OWL, -7.4% with OpenCyc and -14% with DOLCE). F-measure improves with SUMO-OWL and OpenCyc, and degrades with DOLCE.

In the next sections we discuss the methodology and measures we used (Section 4.1), the ontologies we matched and the tests we run (Section 4.2), and the results we obtained (Section 4.3).

4.1 Measures and Methodology

As indicators for measuring how good an alignment is, we used precision, recall and F-measure adapted for ontology alignment evaluation [9].

Precision is defined as the number of correctly found correspondences with respect to a reference alignment (true positives) divided by the total number of found correspondences (true positives and false positives) and recall is defined as the number of correctly found correspondences (true positives) divided by the total number of expected correspondences (true positives and false negatives). A perfect precision score of 1.0 means that every correspondence computed by the algorithm was correct (correctness), whereas a perfect recall score of 1.0 means that all correct correspondences were found (completeness).

To compute precision and recall, the alignment a returned by the algorithm is compared to a reference alignment r.

Precision is given by the formula

$$P(a, r) = \frac{\mid r \cap a \mid}{\mid a \mid}$$

whereas recall is defined as

$$R(a, r) = \frac{\mid r \cap a \mid}{\mid r \mid}$$

We also use the harmonic mean of precision and recall, namely F-measure:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

For carrying out our tests we followed a methodology aimed at ensuring reproducibility of our experiments by reusing existing ontologies and APIs. It may be summarised in the following steps:

1) By exploiting the SWOOGLE Semantic Web Search Engine, http://swoogle.umbc.edu/, we chose 17 ontologies

available online and described in Appendix B. One of them has been reduced by hand to make it more tractable. The other 16 ones are exactly those downloaded from the Web. Our test ontologies consist of 112 concepts on average. This dimension is greater than that of the benchmark ontology used in the OAEI, 2008 (http://oaei.ontologymatching.org/2008/benchmarks/), which contains 33 concepts only.

2) We chose 3 upper ontologies to use in our tests: SUMO (in its restricted OWL version, SUMO-OWL), Cyc (in its open version, OpenCyc) and DOLCE. We downloaded the OWL versions of OpenCyc and DOLCE from http://www.cyc.com/2004/06/04/cyc and http://www.loa-cnr.it/ontologies/DLP_397.owl respectively. The SUO-KIF implementation of SUMO was retrieved from http://www.ontologyportal.org. Our last access dates back to January, 15th, 2009. We performed a translation of SUMO from SUO-KIF into OWL using Sigma [32]. SUMO-OWL contains restricted versions of SUMO, MILO, and all the domain ontologies except the terrorism and airport ones. SUMO-OWL is an order of magnitude smaller than the original SUMO represented in KIF, if one just counts terms, and it is even smaller if one looks at all the axioms necessarily lost in any translation from KIF to OWL.

3) Using the JENA parser, we pre-processed the 17 chosen ontologies and the 3 upper ontologies to remove property restrictions, individuals, and properties.

4) We analysed the ontologies used in our tests in terms of total, simple and composite concepts, concepts including the most common English suffixes and prefixes, "top-level" concepts, ontology maximum depth, stems, and amount of stems that also belong to upper ontologies.

5) We designed 10 tests to run, each consisting of two ontologies to match. For each test we created a reference alignment entirely by hand, consulting dictionaries when we were not sure of the exact meaning of terms. The resulting reference alignments include only and all the correspondences entailed by a reasoning process performed by a human being with a good knowledge of the domain. Although this reasoning process may lead to some subjective choices, its accuracy cannot be obtained in any other way. All the reference alignments are available at http://www.disi.unige.it/person/MascardiV/Software/OntologyMatchingViaUpperOntology.html.

6) For each test, we run the following algorithms: direct alignment without exploiting structure ($parallel\_match$); direct alignment exploiting structure ($structural\_parallel\_match$); alignment via SUMO-OWL, OpenCyc, and DOLCE without exploiting structure (uo_match); alignment via SUMO-OWL, OpenCyc, and DOLCE exploiting structure ($structural\_uo\_match$); mixed alignment obtained by aggregating the structural alignment via SUMO-OWL, OpenCyc, and DOLCE and the structural direct one ($mixed\_match$). The computer where we run our tests is a HP Pavillon Notebook with Intel Core Duo T2250 processor, 1.73 GHz of clock, 2 MB of Level 2 cache, FSB 533 MHz, 1 GB of RAM, and Windows XP.

7) For each test and for each algorithm run within the test, we computed: number of correspondences found by the algorithm; number of correct correspondences found by the algorithm; precision; recall; F-measure; execution time. We exploited the library of evaluators provided by the Alignment API 3.1 for computing the first five values.

8) We aggregated the obtained results by identifying, for each test, the algorithms that give the best precision, recall, and F-measure and by computing the average advantage of using upper ontologies (namely, the average difference between the measures obtained by matching ontologies via upper ontologies and those obtained by performing the direct alignment).

9) From the obtained results we mined the rules of thumb stating when matching via upper ontologies is feasible, when useful, and when useless depending on the features of the matched ontologies.

4.2 Matched Ontologies and Tests

The 17 ontologies used in our tests are described in Appendix B. Tables 1, 2, and 3 summarise the features that best characterise them.

Table 1 shows the results of our analysis of ontologies "as they are". We divided concepts belonging to an ontology (**TotC**) into simple (**SC**) and composite (**CC**) depending on their structure: for example, Classification is a simple concept whereas GeographicArea, Job_title, Politics-The-Diet are composite. We computed the number of top concepts of each ontology, meant as the number of concepts just below *Thing*, and the maximum depth of the tree induced by the *subClassOf* relation. When there was only one concept below *Thing*, we also counted the number of concepts at the second level. This is highlighted by a number written as $1 + n$ in column **Top**, where 1 is the only concept below *Thing*, and n are the concepts below it. We counted the number of concepts containing the most common English prefixes (**Pre**) and suffixes (**Suf**), as defined by [12]. We only considered prefixes and suffixes with at least three letters.

Table 2 shows the results of our analysis based on natural language processing techniques. For each ontology used in our tests, as well as for the three upper ontologies, we created a bag of unique stemmed words in the following way:

1) We automatically tokenised composite concepts obtaining the list of simple words within them. We did not take collocations, namely sequences of words or terms which co-occur more often than would be expected by chance, into account.

2) We manually corrected tokenisation errors in the 17 test ontologies (for example, the automatic tokeniser divided *PSMevaluation* in Ka into *PS* and *Mevaluation*, instead of into *PSM* and *evaluation*). Due to their dimension, we could not perform this manual revision on upper ontologies.

3) We removed stop words from the bags of words obtained after completing steps 1 and 2. We used the 319 stop words defined by the Information Retrieval Research Group of Glasgow University, http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words.

4) We applied Porter's algorithm [33] to remove the commoner morphological and inflectional endings from each word in each bag, thus obtaining bags of unique stemmed words for the 17 test ontologies and the 3 upper ones.

Column **Stems** of Table 2 shows the number of stems for each test ontology; column **S-OWL** shows the number of

Table 1
Quantitative analysis of raw data

Onto.	TotC	SC	CC	Top (%)	Depth (%)	Pre (%)	Suf (%)
Agent	130	61 (47%)	69 (53%)	6 (5%)	7 (5%)	5 (4%)	33 (25%)
Bibtex	15	8 (53%)	7 (47%)	1+14 (7%–100%)	2 (13%)	1 (7%)	2 (13%)
Bios.	88	64 (73%)	24 (27%)	2 (2%)	5 (6%)	0 (0%)	9 (10%)
Eco.	157	46 (29%)	111 (71%)	1+5 (0%–4%)	8 (5%)	1 (0%)	49 (31%)
Food	139	37 (27%)	102 (73%)	1+4 (1%–4%)	5 (4%)	2 (1%)	3 (2%)
Geo.	85	36 (42%)	49 (58%)	5 (6%)	4 (5%)	3 (3%)	28 (33%)
HL7.	60	30 (50%)	30 (50%)	6 (10%)	3 (5%)	3 (5%)	10 (17%)
Ka	96	36 (38%)	60 (62%)	1+7 (1%–8%)	6 (6%)	3 (3%)	33 (34%)
MPEG7	349	240 (69%)	109 (31%)	1+7 (0%–2%)	5 (1%)	8 (2%)	49 (14%)
Rest.	164	21 (13%)	143 (87%)	17 (10%)	3 (2%)	9 (5%)	17 (10%)
Resume	167	106 (63%)	61 (37%)	19 (11%)	3 (2%)	3 (2%)	51 (31%)
Space	165	103 (62%)	62 (38%)	1+3 (1%–2%)	6 (4%)	5 (3%)	26 (16%)
Subj.	171	55 (32%)	116 (68%)	8 (5%)	4 (2%)	3 (2%)	22 (13%)
Topbio	65	15 (23%)	50 (77%)	1+2 (2%–5%)	6 (9%)	5 (8%)	33 (50%)
Travel	84	50 (60%)	34 (40%)	1+2 (1%–4%)	6 (7%)	1 (1%)	16 (19%)
Vac.	32	14 (44%)	18 (56%)	3 (9%)	3 (9%)	1 (3%)	4 (12%)
Vert.	20	14 (70%)	6 (30%)	20 (100%)	1 (5%)	0 (0%)	0 (0%)

Table 2
Quantitative analysis of stemmed concepts

Onto.	Stems	S-OWL	S-OWL%	OCyc	OCyc%	DOL	DOL%
Agent	135	90	67%	123	91%	13	10%
Bibtex	15	3	20%	6	40%	0	0%
Bios.	83	40	48%	61	73%	0	0%
Eco.	129	82	64%	108	84%	25	19%
Food	94	40	43%	77	82%	5	5%
Geo.	104	84	81%	97	93%	12	12%
HL7.	69	52	75%	65	94%	8	12%
Ka	100	59	59%	84	84%	17	17%
MPEG7	331	151	46%	276	83%	22	7%
Rest.	183	88	48%	166	91%	6	3%
Resume	157	98	62%	139	89%	24	15%
Space	177	118	67%	167	94%	9	5%
Subj.	185	109	59%	151	82%	20	11%
Topbio	60	34	57%	51	85%	17	28%
Travel	71	45	63%	66	93%	8	11%
Vac.	25	17	68%	22	88%	4	16%
Vert.	26	17	65%	24	92%	1	4%

common stems between the test ontology and SUMO-OWL, and column **S-OWL%** shows the ratio of common stems with respect to the total number of stems of the test ontology (S-OWL/Stems). The last four columns have the same meaning as the second and third ones, but OpenCyc and DOLCE are considered.

Table 3 provides an objective description of the ontology domains. For extracting information on the domains in an automatic way, we tried three different approaches: 1) a statistical analysis of occurrence of stems in the bag of unique stemmed words, 2) the exploitation of WordNet Domains [4] associated with ontology stems, and 3) the analysis of concepts at the top level (or at the second level, if the top one contained only one concept).

The frequency of stems in the bag of unique stemmed words was significant in some cases (the most frequent stems in Vacation are "vacation" and "relax" which correctly describe the ontology purpose), but failed in other cases (each stem in the bag associated with Bibtex appears only once, thus giving a flat frequency).

The exploitation of WordNet Domains failed in most cases. Many stems are associated with the *factotum* domain and many others are associated with two or three semantically heterogeneous domains. Also, the amount of WordNet Domains that tag stems of an ontology are often too many to give a useful hint on the prevalent ontology domains. A concrete example comes from Ka, which models the knowledge acquisition community (its researchers, topics, products): 30 stems were tagged as belonging to *factotum*; 7 stems were tagged with two domains; 2 stems were tagged with three domains; the domains associated with stems are more than 30 including *art*, *photography*, *religion*, *sport*, *town_planning*. Some domains are associated with more than one stem, thus suggesting that the domain is more important than others, but this is not enough for understanding what the ontology is about. The domain *person* occurs 8 times, whereas *pedagogy* occurs 5 times, *industry* 4, *university* and *publishing* 3. The stem "knowledg", one among those that best characterises Ka, is tagged with *psychological_features* that occurs only 2 times and does not contribute to understanding Ka's domain.

Since implementing a reliable algorithm for automatic extraction of the ontology domain was far from the scope of this paper, the failures we experienced with the first two methods led us to use top concepts for describing ontologies from a qualitative perspective. This approach, which was the simplest to implement, gave satisfactory results in many cases, although it failed for Top-bio, Travel, and Vacation ontologies whose top concepts are meaningless. We feel that the combination of the three approaches should lead to better results.

Finally, Table 4 describes the tests we run. For each test, we show the two matched ontologies (**o** and **o'**), the stems they have in common (**CS**), and the ratio between common stems and total number of different stems (**CS / (St(o) + St(o') - CS)**).

Table 3
Qualitative analysis of the ontology domain

Onto.	Concepts at the top level(s)
Agent	Activity, Agent, Degree, Gender, Job_title, Language
Bibtex	Entry + (Article, Book, Booklet, Conference, Inbook, Incollection, Inproceedings, Manual, Mastersthesis, Misc, Phdthesis, Proceedings, Techreport, Unpublished)
Bios.	LivingThing, MarineAnimal
Eco.	EcologicalConcepts + (EcologicalEntity, EcologicalEnvironment, EcologicalProcess, EcologicalTerminology, EcologicalTrait)
Food	ConsumableThing + (EdibleThing, PotableLiquid, MealCourse, Meal)
Geo.	Classification, GeographicArea, InstallationType, Location, Status
HL7.	RBAC_Reference_Model_Elements, Human, Industry, Organizational_Resources, Organization, Organizational_Tax_Category
Ka	Object + (Event, Organization, Person, Product, Project, Publication, ResearchTopic)
MPEG7	MPEG7Genre + (Animations-special_effect, Drama, Enrichment, Entertainment, Information, Movies, Music)
Rest.	Atmosphere, Beverage, Catering, Cuisine, DatePeriod, Dish, Facility, Meal, OpeningPeriod, Rating, Restaurant, RestaurantGroupParty, RestaurantSeating, Review, Show, SpecialFeature, TimePeriod
Resume	Accomplishment, Address, Award, Career, ContactInfo, Degree, Education, Experience, ExpertiseArea, Industry, Knowledge, Name, Organization, Patents, Person, Publication, Resume, Title, ValuePartition
Space	Spatial_entity + (Geographical_feature, Geopolitical_entity, Place)
Subj.	AppliedSciences, ArtsAndHumanities, Business, History, InterdisciplinaryStudies, Law, Sciences, SocialSciences
Topbio	Domain_entity + (Refining_entity, Self_standing_entity)
Travel	MetaObject + (Domain, DomainIndependent)
Vac.	CBR_DESCRIPTION, CBR_INDEX, CBRCASE
Vert.	Blood, Body_proper, Brain, Bronchus, Cardiac_valve, Half_heart, Head, Heart, Laterality_selector_value, Left_laterality_value, Limb, Liver, Lobe, Lung, Neck, Pericardium, Right_laterality_value, Stomach, Trachea, Trunk

Table 4
Tests run

Test	o	o'	CS	CS / (St(o) + St(o') - CS)
1	Ka	Bibtex	3	3%
2	Biosphere	Top-bio	0	0%
3	Space	Geofile	29	12%
4	Restaurant	Food	10	4%
5	MPEG7	Subject	49	10%
6	Travel	Vacation	4	4%
7	Resume	Agent	26	10%
8	Resume	HL7_RBAC	28	14%
9	Ecology	Top-bio	12	7%
10	Vertebrate	Top-bio	3	4%

4.3 Experimental results

In this section we provide a synthesis of the results of our tests and we discuss them. The complete results can be found in Appendix C.

Table 5 shows the algorithms that gave the best precision, recall, and F-measure for each test, together with their values (in round brackets). S-OWL stands for SUMO-OWL and OCyc for OpenCyc. *Dir* stands for "Direct alignment", *NS* means "No Structure", *S* means "with Structure", and *M* means "Mixed". Sometimes, different algorithms led to the same result. In those cases we listed them all. As far as recall is concerned, we show both the results obtained by methods different from the mixed one (**Best rec. (no mixed)**) and those obtained by all methods, including mixed one (**Best rec. (mixed)**). Since mixed method outperforms all the other ones, the results shown in column **Best rec. (mixed)** always refer to it and we drop the "M" for readability.

Table 5
Algorithm that gives the best precision, recall, F-measure

	Best prec.	Best rec. (no mixed)	Best rec. (mixed)	Best F-meas.
1	OCyc, NS (**0.71**)	S-OWL, S (**0.12**)	S-OWL (**0.13**)	S-OWL, S (**0.18**)
2	Dir, NS & S (**0.23**)	Dir, NS & S; DOLCE, NS & S; S-OWL, S (**0.01**)	S-OWL; DOLCE (**0.02**)	S-OWL, M; DOLCE, M (**0.04**)
3	OCyc, NS (**0.55**)	Dir, NS & S (**0.07**)	S-OWL; OCyc; DOLCE (**0.10**)	OCyc, M (**0.13**)
4	OCyc, NS (**0.42**)	OCyc, S (**0.08**)	OCyc (**0.08**)	OCyc, S & M (**0.12**)
5	OCyc, NS (**0.47**)	S-OWL, S (**0.17**)	OCyc; DOLCE (**0.23**)	S-OWL, NS; OCyc, NS (**0.22**)
6	Dir, NS & S (**0.38**)	Dir, NS & S (**0.07**)	S-OWL; DOLCE (**0.09**)	S-OWL, M (**0.14**)
7	S-OWL, NS (**0.44**)	Dir, S; OCyc, S (**0.06**)	OCyc (**0.10**)	OCyc, M (**0.14**)
8	OCyc, NS (**0.71**)	Dir, S (**0.16**)	S-OWL (**0.22**)	OCyc, NS (**0.20**)
9	Dir, NS & S (**0.32**)	Dir, NS & S; S-OWL, S (**0.09**)	S-OWL (**0.15**)	Dir, NS & S; OCyc, M (**0.14**)
10	OCyc, NS (**0.67**)	OCyc, S (**0.47**)	OCyc (**0.47**)	OCyc, S (**0.24**)

Table 6 synthesises the average advantage in using upper ontologies vs. not using them. Each row refers to the comparison of results obtained by using a given upper ontology (first element of the row's name) and a given method (second element of the row's name: again, **NS** means "No Structure", **S** means "with Structure", and **M** means "Mixed"), and those obtained by performing the direct alignment with the same method.

For example, cell (**(S-OWL, NS)**, **Precision**) reports the average difference in precision between aligning ontologies using SUMO-OWL and performing the direct alignment, both without exploiting the ontology structure. If we identify the precision obtained using SUMO-OWL without structure in experiment i with *p(S-OWL, NS, i)*, and the precision obtained by performing the direct alignment without structure in experiment i with *p(Dir, NS, i)*, then

(**S-OWL, NS**, **Precision**) =
$\Sigma_{i=1}^{10}$ *(p(S-OWL, NS, i) - p(Dir, NS, i)) / 10*

When the mixed method is used, we compared its results

Table 6
Average advantage in using upper ontologies

	Precision	Recall	F-measure
S-OWL, NS	**0.097**	**-0.012** [-0.013]	**-0.012** [-0.011]
OCyc, NS	**0.160** [0.157]	**-0.018** [0.019]	**-0.017** [-0.018]
DOLCE, NS	**-0.131** [-0.145]	**-0.025** [-0.027]	**-0.046** [-0.049]
S-OWL, S	**-0.018** [-0.021]	**0.028**	**0.003** [0.000]
OCyc, S	**-0.017** [-0.004]	**0.029** [0.030]	**0.005** [0.014]
DOLCE, S	**-0.140** [-0.162]	**-0.021** [-0.031]	**-0.045** [-0.053]
S-OWL, M	**-0.077** [-0.089]	**0.066**	**0.014** [0.016]
OCyc, M	**-0.074** [-0.070]	**0.071** [0.070]	**0.022** [0.024]
DOLCE, M	**-0.140** [-0.158]	**0.025** [0.020]	**-0.018** [-0.023]

with those of the direct alignment that gives the best F-measure on that test.

We run the same experiments by exploiting both WordNet 3.0 and WordNet 2.0 as external resource. In square brackets we report the results obtained with WordNet 2.0, if different from those obtained with WordNet 3.0. The improvement with using WordNet 3.0 is negligible.

In order to explain the results we obtained, we carried out a systematic analysis of the relationships among ontology features, methods used, and obtained results, and we mined rules of thumb when possible. We only compared "pure direct" methods and "pure upper ontology-based" ones (second column of Table 5) when we analysed the relationships between recall and ontology features. For sake of readability, in the sequel *uo* stands for *upper ontology-based*.

Structural, non structural and mixed uo methods. As far as precision is concerned, when non structural uo methods succeed (in 70% of our tests) they ensure a significant precision (at least 42% in test 4, but the average on the 7 successful tests is 57%). When they fail, the precision of the succeeding direct matching is not as good: 38% at most, 31% on average. The mixed method outperforms all the other ones when we look at recall but, if we only look at the second column of Table 5, we note that the best recall is always obtained by methods that exploit structure. However the precision of structural methods is greater or equal than that of non structural ones only in tests 2, 6, 9.

By exploiting structure, many correct correspondences that were not found by non structural methods can be retrieved and recall improves. Unfortunately, the same holds for wrong correspondences: those that were not retrieved by non structural methods, may be found when looking at the ontology structure. If we think of wrong correspondences as noise, the exploitation of structure causes a lot of noise which lowers its precision.

Similar considerations explain the results of the mixed method. As far as recall is concerned, the mixed method outperforms all the other ones (third column of Table 5) because it returns all correct correspondences that can be found with both direct and structural uo methods. Besides returning them, however, it also returns all the wrong correspondences and this easily explains its low precision.

Mined rule: *if precision is more important than recall then choose a non structural uo method, otherwise choose either one structural uo method or the mixed one.*

Ontology dimension. Since the number of concepts of our ontologies ranges from 15 (Bibtex) to 349 (MPEG7), we divided the interval [15, 349] into three sub-intervals with the same dimension in order to categorise our ontologies into *small* (less than 127 concepts), *medium* (from 127 to 238 concepts), and *large* (more than 239 concepts). Bibtex, Biosphere, Geofile, HL7_RBAC, Ka, Top-bio, Travel, Vacation and Vertebrate are small. MPEG7 is large. All the remaining ones are medium.

Tests that involve at least one small ontology (1, 2, 3, 6, 8, 9, 10) give very heterogeneous results. In 3 of them the best precision is obtained by structural and non structural direct methods, in the remaining ones by uo methods. Similar considerations on the heterogeneity of results hold for recall.

When no small ontologies are involved (tests 4, 5, 7), results are more coherent: uo methods give the best precision (OpenCyc NS in two tests, SUMO-OWL NS in one) and the best recall (in test 7 the direct structural method has the same performance as OpenCyc, S). The best F-measure is obtained with OpenCyc and SUMO-OWL.

Mined rule: *when at least one matched ontology is small, no rule can be mined; when no small ontologies are involved in the test, uo methods perform better.*

Simple and composite terms. We divided the test ontologies into three broad categories according to the amount of composite and simple terms: *balanced*, where both simple and composite terms are in the range [40%, 60%], *composite*, where composite terms are more than 60%, and *simple*, where simple terms are more than 60%. Agent, Bibtex, Geofile, HL7_RBAC, Travel, and Vacation are balanced; Ecology, Food, Ka, Restaurant, Subject, and Top-bio are composite; Biosphere, MPEG7, Resume, Space, Vertebrate are simple. In tests that involve one balanced and one unbalanced ontology (1, 3, 7, 8), the best precision is always obtained by non structural uo methods (OpenCyc in tests 1, 3, 8; SUMO-OWL in test 7), whereas the best recall is obtained by SUMO-OWL with structure in test 1, by a direct method in tests 3 and 8, and by both uo and direct methods in test 7. In the only test that involves two balanced ontologies, test 6, the best precision and recall are obtained by direct methods (both with and without structure). Tests that involve simple-composite and composite-composite ontologies give heterogeneous results: the best precision is given by direct methods in some tests (2, 9) and by uo methods in other tests (4, 5, 10). The recall is the same either using one uo method or a direct one in tests 2 and 9, whereas it is higher when a uo method is used in tests 4, 5, and 10.

Mined rule: *non structural uo methods give the best precision when one balanced and one unbalanced ontology are matched.*

Common English suffixes and prefixes. We divide ontologies into two balanced categories: those where concepts containing common prefixes and suffixes are strictly less than 20% (Biosphere, Food, MPEG7, Restaurant, Space, Subject, Vacation,

Vertebrate), that we name *poor*, and the other ones (Agent, Bibtex, Ecology, Geofile, HL7_RBAC, Ka, Resume, Top-bio, Travel), that we name *rich*.

In tests that involve both poor ontologies (4, 5), best precision, recall and F-measure are obtained by uo methods. In tests that involve one poor and one rich ontology (2, 3, 6, 10), we have 50% of cases where uo perform better than direct methods as far as both precision and recall is concerned (3, 10).

When two rich ontologies are matched (tests 1, 7, 8, 9), in 75% of cases uo methods give the best precision and recall, but test 9 witnesses a better performance of direct methods, even if we look at F-measure.

Mined rule: *when both matched ontologies are poor, uo methods give better results; in the other cases, no clear rule can be mined.*

Concepts at top-level(s). We divided ontologies into *broad* (those with at least 5 concepts at the top level(s)) and *narrow*. Agent, Bibtex, Ecology, Food, Geofile, HL7_RBAC, Ka, MPEG7, Restaurant, Resume, Subject, Vertebrate are broad; Biosphere, Space, Top-bio, Travel, Vacation are narrow. In tests that involve two broad ontologies (1, 4, 5, 7, 8), uo methods perform better: the best precision is always obtained by uo methods (OpenCyc NS in four cases, SUMO-OWL NS in one); the best recall is obtained by uo methods in three tests, by structural direct method in one test, and by both direct and uo methods in the remaining test. Tests that involve two narrow ontologies (2, 6) show an out-performance of direct methods with respect to precision and recall (although in test 2 the same recall is obtained by uo methods too). Mixed method gives the best F-measure. In the three tests that involve one narrow and one broad ontology (3, 9, 10), heterogeneous results are obtained: the best precision is obtained by OpenCyc NS in tests 3 and 10 and by direct methods in test 9. The best recall is obtained by direct methods in test 3 and 9 (SUMO-OWL S gives the same result) and by OpenCyc S in test 10. The best F-measure is obtained by OpenCyc either mixed in tests 3 and 9 (where direct methods give the same F-measure) or structural in test 10.

Mined rule: *direct methods give a better precision when both matched ontologies are narrow; uo methods give the best precision when two broad ontologies are matched; recall is not strongly influenced by the number of concepts at the top level(s).*

Maximum depth of the ontology. Eleven ontologies, that we name *shallow*, have a maximum depth lower than 6 (Bibtex, Biosphere, Food, Geofile, HL_RBAC, MPEG7, Restaurant, Resume, Subject, Vacation, Vertebrate). The remaining six ontologies, that we name *deep*, have a depth between 6 and 8. Five tests (1, 3, 6, 7, 9) involve at least one deep ontology. In these tests uo methods give the best precision in the 60% of cases (tests 1 and 3 with OpenCyc NS; test 7 with SUMO-OWL NS). Best recall and F-measure are more often obtained by direct methods. In the remaining five tests where both ontologies are shallow the best recall is obtained by a uo method in 60% of cases (tests 4, 5, 10; test 2 gives the same recall when both a uo and a direct method is used) and the best precision is obtained by a uo method in the 80% of cases (tests 4, 5, 8, 10, with OpenCyc NS). The best F-measure is always obtained by a uo method.

Mined rule: *uo methods perform better when matched ontologies are both shallow.*

Analysis of stems. Table 4 shows, for each experiment we run, the number of stems that appear in both matched ontologies (CS, "Common Stems") and the number of different stems appearing in both ontologies (Stems(o) + Stems(o') - CS). We computed the ratio between these values and we observed that, when it is greater than 8% (tests 3, 5, 7, 8), the best results in term of precision and recall are always obtained by uo methods. When $(CS/(Stems(o) + Stems(o') - CS)) \leq 7\%$ no rule can be mined: the best precision is obtained by direct methods in three tests (2, 6, 9) and by uo methods in the other three (1, 4, 10).

The number of stems appearing in a test ontology that also appear in an upper ontology (Table 2) does not influence the results. Only 20% of Bibtex's stems appear in SUMO-OWL, and only 40% appear in OpenCyc. Nevertheless, uo methods perform better than direct ones in test 1 that involves Bibtex. On the other hand, many stems of Travel and Vacation appear in both SUMO-OWL (63% and 68% respectively) and in OpenCyc (93% and 88% respectively), but direct matching between Travel and Vacation (test 6) outperforms uo methods as far as both precision and recall are concerned.

Mined rule: *uo methods perform better when the ratio between the number of common stems and the total number of stems of the matched ontologies is greater than 7%.*

Execution time. Table 7 compares execution time, measured in seconds, of direct methods (structural vs non structural) and of structural, non structural and mixed methods via OpenCyc and SUMO-OWL (considering only the on-line part of the computation). Direct matching methods always perform better than uo ones, but their efficiency is payed by a lower recall and precision in most cases. The alignment a between ontology o and upper ontology uo can be computed and stored once and for all, and can be re-used whenever o must be matched with another ontology o' via uo. Thus, we consider it an off-line activity and it does not contribute to the on-line execution time shown in Table 7. However, we must note that aligning an ontology o from our test set with OpenCyc turned out to be a very time-consuming activity and required more than 6 hours in three cases, more than 4 in two cases, and from 5 minutes to 2 hours in the remaining twelve cases. The average time was **4 hours**. Instead, aligning an ontology o with SUMO-OWL required no more than 20 minutes in all the 17 alignments we performed (the average time was **7 minutes**). In Table 7, execution time of structural and non-structural direct methods refers to the execution time required by $structural\_parallel\_match$ and $parallel\_match$ respectively. Execution time of non structural methods that involve upper ontologies (**SUMO-OWL NS vs OpenCyc NS**) refers to the composition phase only. In the same way, execution time of structural methods via upper ontology (**SUMO-OWL**

S vs OpenCyc S) refers to the time required by executing $structural\_compose$ on pre-computed alignments $o-uo$ and $o'-uo$. Finally, execution time of mixed methods (**SUMO-OWL M vs OpenCyc M**) refers to the aggregation stage only. If we only consider on-line activities, OpenCyc is more efficient than SUMO-OWL when non-structural and mixed methods are considered. However, if we consider execution time of both on-line and off-line activities, methods that use OpenCyc are about 35 times more time-consuming than those based on SUMO-OWL.

Mined rule: *In those situations where the developer might accept less quality but gain more performance with respect to the time (considering both on-line and off-line activities), SUMO-OWL is the best choice. OpenCyc ensure higher quality, but may require hours for performing the off-line alignments.*

Table 7
On-line activities: best execution time in seconds

	Dir S vs dir NS	**S-OWL NS vs OCyc NS**	**S-OWL S vs OCyc S**	**S-OWL M vs OCyc M**
1	NS (**13.89**)	OCyc (**0.19**)	S-OWL (**3.27**)	OCyc (**0.16**)
2	NS (**5.00**)	OCyc (**0.31**)	S-OWL (**4.78**)	OCyc (**0.13**)
3	NS (**10.85**)	OCyc (**0.39**)	S-OWL (**12.48**)	OCyc (**0.69**)
4	NS (**30.94**)	OCyc (**0.34**)	S-OWL (**11.53**)	OCyc (**0.38**)
5	NS (**58.56**)	OCyc (**0.48**)	S-OWL (**36.13**)	OCyc (**0.94**)
6	NS (**6.99**)	OCyc (**0.31**)	S-OWL (**8.19**)	OCyc (**0.23**)
7	NS (**29.99**)	OCyc (**0.39**)	S-OWL (**34.52**)	OCyc (**0.55**)
8	NS (**12.65**)	OCyc (**0.27**)	S-OWL (**7.16**)	OCyc (**0.50**)
9	NS (**14.47**)	OCyc (**0.38**)	S-OWL (**8.41**)	OCyc (**0.34**)
10	NS (**3.94**)	S-OWL (**0.25**)	S-OWL (**3.42**)	OCyc (**0.16**)

5 Conclusions

In this paper we described a set of algorithms for exploiting upper ontologies as bridges in the ontology matching process and we discussed the results of our experiments.

Someone might argue that DOLCE is the only "pure upper ontology" we used, since both SUMO-OWL and OpenCyc are large-coverage general-purpose ontologies that include many domain-specific concepts in addition to the upper-level ones. A "pure upper ontology" would by definition fail to cover most of the concepts present in a domain ontology, and DOLCE demonstrates that this is the case.

The results of our tests may be summarised in the following way:

1) In 70% of our experiments, the best precision was obtained by methods based on upper ontologies. The best recall is always obtained by the mixed method. The best F-measure is always obtained by methods that exploit upper ontologies.
2) Methods based on upper ontologies are always less efficient than direct ones. If we consider both on-line and off-line activities, methods that use OpenCyc are always less efficient than methods that exploit SUMO-OWL.
3) Using WordNet 3.0 instead of WordNet 2.0 gives a negligible advantage.
4) The gap between a "pure" and foundational upper ontology like DOLCE and the test ontologies, clearly shown in Table 2, is too large to make DOLCE a suitable bridge between test ontologies.
5) OpenCyc and SUMO-OWL are large and detailed enough and give comparable results, although OpenCyc performs slightly better than SUMO-OWL.

The main future direction of our work is to improve the recall of our matching methods. Obtaining a precision higher than the recall is a typical result of any automatic ontology matching system. The results of the OAEI competitions confirm this claim. For example, the average recall obtained by the 7 matching systems that participated to the OAEI-2008 competition on web directories data-set, http://www.disi.unitn.it/~pane/OAEI/2008/directory/result/, was 30% and the best one was 41%. These results give a measure of the intrinsic complexity of obtaining high recall values, despite the used matching system. The best recall we obtained in our experiments is 47% but the average is 16%, if we only consider the best results shown in Table 5. Some reasons explaining the low average recall of our tests were empirically mined from the systematic analysis carried out in Section 4.3. Another explanation is that the ontologies involved in our test set are definitely larger than those used for the OAEI-2008 competitions: the largest ontology from the OAEI-2008 web directories data-set contains 16 concepts and the benchmark OAEI-2008 ontology, http://oaei.ontologymatching.org/2008/benchmarks/, contains 33 concepts. The largest ontology in our test set, MPEG7, contains 349 concepts. Also, our ontologies were neither adapted nor reduced (apart form one) for the purpose of running our experiments. Identifying the correct correspondences to be included in the reference alignments, developed totally by hand to achieve the highest accuracy, was an hard task even for a human being and it turned out to be even harder for our automatic matching algorithms.

To overcome the limitations of our approach we are planning to consider more expressive relations, to extend our algorithms in order to properly deal with correspondences between individuals and properties, and to extend structural matching methods in order to explore a more significant portion of the hierarchy. Structural alignment might be extended by taking advantage of properties and their arguments, links among rules, links other than super/subclass structure. Taking the comments in the OWL description into account would also help. We are also interested in studying how much an automatic algorithm would speed up the matching process, and quantifying the speed-up over manual matching. Quantitative aspects concerning time and efficiency are often neglected by the existing literature, but experiments in this direction would be helpful for the ontology matching community. Studying methods for extracting the ontology domain in an automatic way is a challenging activity that we are going to start.

Acknowledgements

We sincerely thank J. Euzenat and P. Shvaiko that granted permission to us to adapt the definitions and the figures of their book [10]. We are grateful to A. Pease that provided us with the OWL translation of SUMO, and to him and G. de Melo that suggested some future directions of our work.

Finally, we thank D. Buscaldi for his suggestions on WordNet domains and the anonymous reviewers for their constructive comments and useful advices.

The work of the first, second and third authors was partly supported by the Italian research project Iniziativa Software CINI-FinMeccanica, the EU 6 FP KP-Lab project, and the Spanish research project Text-Mess (CICYT TIN2006-15265-C06-04) respectively.

REFERENCES

[1] Z. Aleksovski, M. C. A. Klein, W. ten Kate, and F. van Harmelen. Matching unstructured vocabularies using a background ontology. In *Proc. of EKAW 2006*, pages 182–197. Springer, 2006.

[2] Z. Aleksovski, W. ten Kate, and F. van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In P. Shvaiko, J. Euzenat, N. Noy, H. Stuckenschmidt, R. Benjamins, and M. Uschold, editors, *Proc. of OM-2006*, 2006.

[3] American National Standard. KIF Knowledge Interchange Format – draft proposed American National Standard (dpANS) NCITS.T2/98-004, 1998.

[4] L. Bentivogli, P. Forner, B. Magnini, and E. Pianta. Revising WordNet domains hierarchy: Semantics, coverage, and balancing. In *Proc. of COLING 2004*, pages 101–108. 2004.

[5] P. A. Bernstein, A. Y. Halevy, and R. Pottinger. A vision of management of complex models. *SIGMOD Record*, 29(4):55–63, 2000.

[6] P. Bouquet, L. Serafini, and S. Zanobini. Peer-to-peer semantic coordination. *Journal of Web Semantics*, 1(2), 2005.

[7] E. Brill, S. Dumais, and M. Banko. An analysis of the askmsr question-answering system. In *Proc. of EMNLP 2002*, 2002.

[8] N. Casellas, M. Blázquez, A. Kiryakov, P. Casanovas, M. Poblet, and R. Benjamins. OPJK into PROTON: Legal domain ontology integration into an upper-level ontology. In R. Meersman and et al., editors, *Proc. of WORM 2005*, pages 846–855. Springer, 2005.

[9] H. H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In A. B. Chaudhri, M. Jeckle, E. Rahm, and R. Unland, editors, *Proc. of NODe 2002*, pages 221–237. Springer, 2002.

[10] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, 2007.

[11] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening ontologies with DOLCE. In A. Gómez-Pérez and V. R. Benjamins, editors, *Proc. of EKAW 2002*, pages 166–181. Springer, 2002.

[12] A. Gillett. Using english for academic purposes – a guide for students in higher education, 2009. School of Combined Studies, University of Hertfordshire, Hatfield, UK.

[13] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In R. Meersman, Z. Tari, M-S. Hacid, J. Mylopoulos, B. Pernici, Ö. Babaoglu, H-A Jacobsen, J. P. Loyall, M. Kifer, and S. Spaccapietra, editors, *Proc. of CoopIS, DOA, and ODBASE 2005*, pages 347–365. Springer, 2005.

[14] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Discovering missing background knowledge in ontology matching. In G. Brewka, S. Coradeschi, A. Perini, and P. Traverso, editors, *Proc. of ECAI 2006*, pages 382–386. IOS Press, 2006.

[15] J. Gracia, V. Lopez, M. d'Aquin, M. Sabou, E. Motta, and E. Mena. Solving semantic ambiguity to improve semantic web based ontology matching. In P. Shvaiko, J. Euzenat, F. Giunchiglia, and B. He, editors, *Proc. of OM-2007*, 2007.

[16] P. Grenon, B. Smith, and L. Goldberg. Biodynamic ontology: Applying BFO in the biomedical domain. In D. M. Pisanelli, editor, *Ontologies in Medicine*, Studies in Health Technology and Informatics, pages 20–38. IOS Press, 2004.

[17] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[18] A. Hameed, A. D. Preece, and D. H. Sleeman. Ontology reconciliation. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 231–250. Springer, 2004.

[19] J. Hammer and D. McLeod. An approach to resolving semantic heterogeneity in a federation of autonomous, heterogeneous database systems. *International Journal of Intelligent and Cooperative Information Systems*, 2(1):51–83, 1993.

[20] J. A. Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, 16(2):30–37, 2001.

[21] H. Herre, B. Heller, P. Burek, R. Hoehndorf, F. Loebe, and H. Michalek. General formal ontology (GFO): A foundational ontology integrating objects and processes. part i: Basic principles. Technical report, Research Group Ontologies in Medicine (Onto-Med), University of Leipzig, 2006. Version 1.0, Onto-Med Report Nr. 8, 01.07.2006.

[22] E. Hovy. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In *Proc. of LREC'98*, pages 535–542, 1998.

[23] J. Kahng and D. McLeod. Dynamic classification ontologies: mediation of information sharing on cooperative federated database systems. In *Cooperative Information Systems: Trends and Directions*, pages 179–203. Academic Press, 1998.

[24] D. Lenat and R. Guha. *Building large knowledge-based systems*. Addison Wesley, 1990.

[25] J. Li. LOM: A lexicon-based ontology mapping tool. In *Proc. of PerMIS'04*, 2004.

[26] V. Mascardi, P. Rosso, and V. Cordì. Enhancing communication inside multi-agent systems – an approach based on alignment via upper ontologies. In *Proc. of MALLOW-AWESOME'007*, pages 92–107, 2007.

[27] G. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[28] P. Mitra, G. Wiederhold, and J. Jannink. Semi-automatic integration of knowledge sources. In *Proc. of FUSION'99*, 1999.

[29] I. Niles and A. Pease. Towards a standard upper ontology. In C. Welty and B. Smith, editors, *Proc. of FOIS 2001*, pages 2–9. ACM Press, 2001.

[30] N. F. Noy and M. A. Musen. SMART: Automated support for ontology merging and alignment. In *Proc. of KAW'99*, 1999.

[31] D. E. O'Leary. Impediments in the use of explicit ontologies for KBS development. *International Journal of Human-Computer Studies*, 46(2-3):327–337, 1997.

[32] A. Pease. The Sigma ontology development environment. In F. Giunchiglia, A. Gomez-Perez, A. Pease, H. Stuckenschmidt, Y. Sure, and S. Willmott, editors, *Proc. of ODS 2003*, 2003.

[33] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[34] C. Reynaud and B. Safar. Exploiting wordnet as background knowledge. In P. Shvaiko, J. Euzenat, F. Giunchiglia, and B. He, editors, *Proc. of OM-2007*, 2007.

[35] M. Sabou, M. d'Aquin, and E. Motta. Using the semantic web as background knowledge for ontology mapping. In P. Shvaiko, J. Euzenat, N. Noy, H. Stuckenschmidt, R. Benjamins, and M. Uschold, editors, *Proc. of OM-2006*, 2006.

[36] S. Shehata, F. Karray, and M. Kamel. Enhancing search engine quality using concept-based text retrieval. In *Web Intelligence*, pages 26–32. IEEE Computer Society, 2007.

[37] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous and autonomous database systems. *ACM Computing Surveys*, 22(3):183–236, 1990.

[38] J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing, 1999.

[39] G. Stoilos, G. B. Stamou, and S. D. Kollias. A string metric for ontology alignment. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *Proc. of ISWC 2005*, pages 624–637. Springer, 2005.

[40] H. Stuckenschmidt, F. van Harmelen, L. Serafini, P. Bouquet, and F. Giunchiglia. Using C-OWL for the alignment and merging of medical ontologies. In U. Hahn, editor, *Proc of KRMed'04*, pages 88–101, 2004.

[41] W. R. van Hage, S. Katrenko, and G. Schreiber. A method to combine linguistic ontology-mapping techniques. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *Proc. of ISWC 2005*, pages 732–744. Springer, 2005.

[42] P. R. S. Visser, D. M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. An analysis of ontological mismatches: Heterogeneity vs. interoperability. In A. Farquhar and M. Gruninger, editors, *Proc. of 1997 AAAI Spring Symposium*, pages 164–172. AAAI Press, 1997.

[43] P. R. S. Visser, D. M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. Assessing heterogeneity by classifying ontology mismatches. In N. Guarino, editor, *Proc. of FOIS'98*, pages 148–162. IOS Press, 1998.

[44] W3C. OWL Web Ontology Language Overview – W3C Recommendation 10 February 2004, 2004.

[45] W3C. RDF Vocabulary Description Language 1.0: RDF Schema – W3C Recommendation 10 February 2004, 2004.

[46] W3C. RDF/XML Syntax Specification (Revised) – W3C Recommendation 10 February 2004, 2004.

[47] Wikipedia. Upper ontology – Wikipedia, the Free Encyclopedia, 2009. [Online; accessed 15-January-2009].