Ontology Matching

OM-2006

Papers from the ISWC Workshop

Introduction

Ontology matching is a key interoperability enabler for the Semantic Web since it takes the ontologies as input and determines as output correspondences between the semantically related entities of those ontologies. These correspondences can be used for various tasks, such as ontology merging, query answering, data translation, or for navigation on the Semantic Web. Thus, matching ontologies enables knowledge and data expressed in the matched ontologies to interoperate.

The workshop had two goals:

- To bring together academic and industry leaders to assess how academic advances are addressing real world requirements. The workshop strives to improve academic awareness of industrial needs, and therefore, direct research towards those needs. Simultaneously, the workshop serves to inform industry representatives about existing research efforts that may meet their business needs. Moreover, it is central to the aims of the workshop to evaluate how technologies for ontology matching are going to evolve, which research topics are in the academic agenda and how these can fit emerging business issues.

- To conduct an extensive evaluation of ontology matching approaches via the OAEI (Ontology Alignment Evaluation Initiative) 2006 campaign, `http://oaei.ontologymatching.org/2006`. The particular focus of this year's OAEI campaign is on real world matching tasks from specific domains, such as medicine, food. Therefore, the ontology matching evaluation initiative itself provides a solid ground for discussion of how well the current approaches are meeting business needs.

We received 22 submissions for the technical track of the workshop. The program committee selected 6 submissions for oral presentation and 11 submissions for poster presentation. Ten matching systems participated in this year's OAEI campaign. Amit Sheth (University of Georgia and Semagix) gave a keynote address at the workshop. Further information about the Ontology Matching workshop can be found at: `http://om2006.ontologymatching.org/`.

Acknowledgments. We thank all members of the program committee, authors, keynote speaker and local organizers for their efforts. We appreciate support from the Knowledge Web FP6 Network of Excellence.

knowledge**web**
realizing the semantic web

Pavel Shvaiko
Jérôme Euzenat
Natalya Noy
Heiner Stuckenschmidt
Richard Benjamins
Michael Uschold

November 2006

Organization

Organizing Committee

Pavel Shvaiko, University of Trento, Italy
Jérôme Euzenat, INRIA Rhône-Alpes, France
Natalya Noy, Stanford University, USA
Heiner Stuckenschmidt, University of Mannheim, Germany
Richard Benjamins, iSOCO, Spain
Michael Uschold, The Boeing Company, USA

Program Committee

Benjamin Ashpole, Bashpole, Inc., USA
Richard Benjamins, iSOCO, Spain
Paolo Bouquet, University of Trento, Italy
Jérôme Euzenat, INRIA Rhône-Alpes, France
Fausto Giunchiglia, University of Trento, Italy
Andreas Hess, Vrije Universiteit Amsterdam, Netherlands
Wei Hu, Southeast University, China
Jingshan Huang, University of South Carolina, USA
Todd Hughes, DARPA/IXO, USA
Michael Huhns, University of South Carolina, USA
Ryutaro Ichise, National Institute of Informatics, Japan
Yannis Kalfoglou, University of Southampton, UK
Deborah McGuinness, Stanford University, USA
Meenakshi Nagarajan, University of Georgia, USA
Natalya Noy, Stanford University, USA
Satya Sahoo, University of Georgia, USA
Marco Schorlemmer, IIIA-CSIC, Spain
Pavel Shvaiko, University of Trento, Italy
Steffen Staab, University of Koblenz, Germany
Umberto Straccia, ISTI-C.N.R., Italy
Heiner Stuckenschmidt, University of Mannheim, Germany
York Sure, University of Karlsruhe, Germany
Michael Uschold, The Boeing Company, USA
Petko Valtchev, University of Montreal, Canada
Mikalai Yatskevich, University of Trento, Italy

Table of Contents

PART 1 - Technical Papers

Using the Semantic Web as Background Knowledge for Ontology Mapping
Marta Sabou, Mathieu d'Aquin, Enrico Motta 1

Exploiting the Structure of Background Knowledge Used in Ontology Matching
Zharko Aleksovski, Warner ten Kate, Frank van Harmelen 13

Towards Understanding the Needs of Cognitive Support for Ontology Mapping
Sean M. Falconer, Natalya F. Noy, Margaret-Anne Storey 25

Applying an Analytic Method for Matching Approach Selection
Malgorzata Mochol, Anja Jentzsch, Jérôme Euzenat 37

Arguing over Ontology Alignments
Loredana Laera, Valentina Tamma, Jérôme Euzenat, Trevor Bench-Capon, Terry Payne 49

Improving Automatically Created Mappings Using Logical Reasoning
Christian Meilicke, Heiner Stuckenschmidt, Andrei Tamilin 61

PART 2 - OAEI Papers

Results of the Ontology Alignment Evaluation Initiative 2006
Jérôme Euzenat, Malgorzata Mochol, Pavel Shvaiko, Heiner Stuckenschmidt, Ondřej Šváb, Vojtěch Svátek, Willem Robert van Hage, Mikalai Yatskevich 73

AUTOMS: Automated Ontology Mapping through Synthesis of Methods
Konstantinos Kotis, Alexandros Valarakos, George Vouros 96

COMA++: Results for the Ontology Alignment Contest OAEI 2006
Sabine Massmann, Daniel Engmann, Erhard Rahm 107

DSSim-ontology Mapping with Uncertainty
Miklos Nagy, Maria Vargas-Vera, Enrico Motta 115

The Results of Falcon-AO in the OAEI 2006 Campaign
Wei Hu, Gong Cheng, Dongdong Zheng, Xinyu Zhong, Yuzhong Qu 124

Results of the HMatch Ontology Matchmaker in OAEI 2006
Silvana Castano, Alfio Ferrara, Gianpaolo Messa 134

JHU/APL Onto-Mapology Results for OAEI 2006
Wayne L. Bethea, Clayton R. Fink, John S. Beecher-Deighan 144

NLM Anatomical Ontology Alignment System. Results of the 2006 Ontology Alignment Contest
Songmao Zhang, Olivier Bodenreider 153

OWL-CtxMatch in the OAEI 2006 alignment contest
Slawomir Niedbala 165

PRIOR System: Results for OAEI 2006
Ming Mao, Yefei Peng 173

Result of Ontology Alignment with RiMOM at OAEI'06
Yi Li, Juanzi Li, Duo Zhang, Jie Tang 181

PART 3 - Posters

When Usual Structural Alignment Techniques Don't Apply
Chantal Reynaud, Brigitte Safar 191

A Framework for Combining Ontology and Schema Matchers with Dempster-Shafer
Paolo Besana 196

Aligning Multiple Anatomical Ontologies Through a Reference
Songmao Zhang, Olivier Bodenreider 201

Combining Ontology Mapping Methods Using Bayesian Networks
Ondřej Šváb, Vojtěch Svátek 206

Discovering and Merging Keyword Senses Using Ontology Matching
Mauricio Espinoza, Raquel Trillo, Jorge Gracia, Eduardo Mena 211

A Peer-to-Peer Based Semantic Agreement Approach for Information Systems Interoperability
I Wayan Simri Wicaksana, Kokou Yétongnon 216

An Ontology Mapping Algorithm between Heterogeneous Product Classification Taxonomies
Wooju Kim, Sangun Park, Siri Bang, Sunghwan Lee 221

Recognizing Emergent Nodes in Aligning Multiple Document Taxonomies
Tim Musgrove 226

FORPM: Boosting Users' Effect on Ontology Matching
Dunwei Wen, Xiaohu Fan, Fuhua Lin 231

OntoMas: a Tutoring System Dedicated to Ontology Matching
Mirella Huza, Mounira Harzallah, Francky Trichet 236

SOMET: Shared Ontology Matching Environment
Liam Magee 241

Using the Semantic Web as Background Knowledge for Ontology Mapping

Marta Sabou, Mathieu d'Aquin, and Enrico Motta

Knowledge Media Institute (KMi)
The Open University, Milton Keynes, United Kingdom
{r.m.sabou, m.daquin, e.motta}@open.ac.uk

Abstract. While current approaches to ontology mapping produce good results by mainly relying on label and structure based similarity measures, there are several cases in which they fail to discover important mappings. In this paper we describe a novel approach to ontology mapping, which is able to avoid this limitation by using background knowledge. Existing approaches relying on background knowledge typically have one or both of two key limitations: 1) they rely on a manually selected reference ontology; 2) they suffer from the noise introduced by the use of semi-structured sources, such as text corpora. Our technique circumvents these limitations by exploiting the increasing amount of semantic resources available online. As a result, there is no need either for a manually selected reference ontology (the relevant ontologies are dynamically selected from an online ontology repository), or for transforming background knowledge in an ontological form. The promising results from experiments on two real life thesauri indicate both that our approach has a high precision and also that it can find mappings, which are typically missed by existing approaches.

Keywords: ontology mapping, background knowledge, semantic web

1 Introduction

Ontology mapping techniques are essential for building semantic bridges between ontologies. However, current approaches suffer from a number of problems. First of all, most approaches do not provide a formal semantics to the mapping structures they produce[1]. As a result it is difficult for reasoners to make use of these structures, e.g., to answer queries across ontologies [9]. More importantly, current approaches to ontology mapping [11, 13] heavily rely on string-based and structure-based similarity measures. While these techniques can produce good results, there are also numerous examples in which they fail to find mappings.

A few approaches [1, 14, 15] have considered the use of external background knowledge as a way to obtain semantic mappings between syntactically dissimilar ontologies, i.e., to overcome the aforementioned limitations. However, obtaining the right background knowledge is problematic. Some approaches rely on richly

---

[1] A notable exception is the CtxMatch/S-Match algorithms (see Section 2.1).

axiomatized domain ontologies [1], but unfortunately such ontologies do not exist in all domains and even when they exist, they are unlikely to cover all the intended mappings between the input ontologies. In addition, there are scenarios where it is not possible to select the relevant ontology in advance. For instance, in Semantic Web applications like PowerAqua [9], the domains of the terms to be mapped cannot be determined a priori and whatever background knowledge is needed, must be identified dynamically and in real-time. To avoid the problems associated with the manual selection of an ontology, other techniques try to derive the required background knowledge from weakly structured textual sources [15]. However, given the current limitations in information extraction technology, they then suffer from the resulting noise.

The recent growth of the Semantic Web has resulted in an increased amount of online available semantic data and has led to the first search engine to exploit this data, Swoogle [5]. Our hypothesis is that ontology mapping, while trying to cope with the heterogeneity of the Semantic Web, could actually exploit it. In other words, online available ontologies could provide the background knowledge sources, which are needed to support ontology mapping and to overcome the problems mentioned above. On the one hand, they can be selected dynamically, thus circumventing the need for an a priori, manual ontology selection. On the other hand, by relying on semantic sources, we avoid the inherent noise caused by information extraction based methods.

In this paper we build on these ideas and we describe an approach to ontology mapping which goes beyond similarity-based algorithms by dynamically locating and using relevant background knowledge (Section 3). Since our method derives mappings between pairs of concepts, it can be used to map semantic structures ranging from shallow thesauri to clearly formalized ontologies. We start by discussing in detail the importance of background knowledge in ontology mapping.

2 Motivation

We discuss two major limitations of current ontology mapping approaches that only rely on label and structure similarity (i.e., syntactic approaches), namely that they don't provide semantic mappings (Section 2.1) and that they fail to discover some correct mappings when the mapped ontologies are syntactically dissimilar (Section 2.2). We then point out that while the use of background knowledge can be a solution to these limitations, existing approaches using such knowledge have their shortcomings (Section 2.3).

2.1 Syntactic Approaches Do Not Provide Semantic Mappings

Semantic Web tools, such as PowerAqua, which wish to reason on the results of mapping techniques require that the discovered mappings are expressed as semantic relations between the entities of the ontologies. Formal mapping languages, such as C-OWL [2], envision a wide range of semantic relations that can hold between the entities of two ontologies (e.g., narrower, disjoint). However, few existing mapping techniques are able to discover such semantic mappings.

An analysis of the state of the art of mapping systems presented in [13] explains to some extent the lack of approaches that can provide semantic mappings. The major factor seems to be that most systems combine a range of non-semantic techniques, such as terminological approaches (exploiting string similarity between labels), structural approaches (relying on the structure of the mapped ontologies), and extensional approaches (mapping concepts on the basis of shared instances). Only few systems rely on semantic techniques (also called model based approaches in [13]), thus exploiting the semantics both of the mapped ontologies, and the mapping language, to infer mappings from the available knowledge. As a result, during the last Ontology Alignment Contest (OAC) [6] only one algorithm (CtxMatch [3]) was able to produce partial semantic mappings in the form of subconcept relations. The other techniques produce confidence based mappings that are derived by aggregating the output of terminological and structural algorithms. Unfortunately, this kind of low semantic (*quantitative*) relations are difficult to interpret and to exploit in reasoning procedures. On the contrary, semantic techniques should produce meaningful relations between the mapped entities, on which further reasoning can be applied. They should focus on *qualitatively* good mappings that can be justified and explained through the knowledge and inferences used to deduce them.

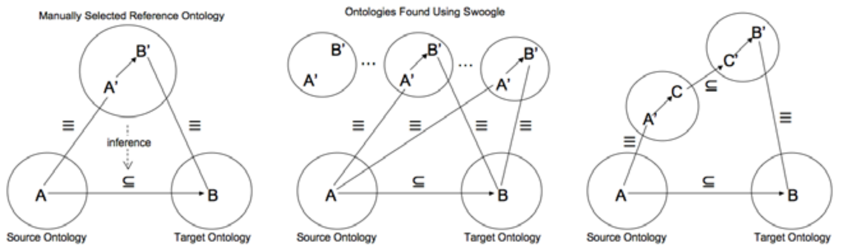
2.2 Syntactic Approaches Fail on Dissimilar Ontologies

As already observed by [1], traditional methods fail when there is little lexical overlap between the labels of the ontology entities, or when the ontologies have weak or dissimilar structures. This observation has been verified to some extent in the last OAC [6]. In the first task of this contest where a base ontology was mapped to its systematically modified versions, the performance of most methods decreased significantly in the test cases where important changes have been performed to the labels and structures of the ontologies (tests 250 - 266). In fact, traditional techniques are based on the hypothesis of an equivalence between some forms of syntactic correspondences and semantic relations. While it is true that, in many cases, string and structural similarities can imply meaningful mappings, this hypothesis is far from being always verified. For instance, the relation between the concepts *Beef* and *Food* may not be discovered on the basis of syntactical considerations, but becomes obvious when considering the meaning of these concepts (their semantics). By ignoring such semantics, syntactic techniques fail to identify several important mappings.

2.3 How is Background Knowledge Currently Used?

The previous sections suggest that the meaning of the mapped concepts should be considered to discover meaningful and syntactically unidentifiable mappings. Unfortunately, while meaning on the Semantic Web is expressed using ontologies, in the case of ontology mapping, the constituents of a mapping can only be given meaning in the context of their own distinct ontology, which cannot cover both the source and target elements, as well as the relation linking them. In other words, a semantic mapping between two ontologies could only be interpreted in a *larger* domain than the ones of these ontologies. Therefore, in order to achieve semantic mapping, the integration of external knowledge is required as a way to cover both input ontologies and to fill the semantic gap between them. So far the following types of background knowledge have been used in mapping:

1. WordNet is one of the most often used sources of background knowledge. For example, CTxMatch [3] (and its follow-up, SMatch [7]) translates ontology labels into logical formulae between their constituents, and maps these constituents to corresponding senses in WordNet. A SAT solver is then used to derive semantic mappings between the different concepts. This approach has been recently extended to handle the problem of *missing background knowledge* [8]. The lack of knowledge is detected and compensated during the mapping process, using techniques still relying on WordNet as a source of knowledge. When using WordNet, it is important to be aware that it is a lexical resource (rather than a truly semantic resource), relating terms by using terminological relations like synonymy or hypernymy. Therefore, it can be seen as a source of linguistic knowledge, useful in relating labels during the terminological step of a matching procedure.



(a) (b) (c)

Fig. 1. Using ontologies as background knowledge for semantic mapping: (a) using a manually selected reference ontology (e.g., [1]); (b) using Swoogle to find the appropriate ontologies (S1) (c) recursively exploiting multiple ontologies (S3).

2. Reference Domain Ontologies. Another approach is to rely on a reference domain ontology as a semantic bridge between two ontologies. In [1], the authors experimentally prove that state of the art matchers fail to satisfactorily match two weakly structured vocabularies of medical terms. As a solution, they propose to use the DICE ontology as a source of background knowledge. Terms from the two vocabularies are first mapped to so called anchor terms in DICE and then their mapping is deduced based on the semantic relation of the anchor terms (see Figure 1(a)). As such, the obtained mappings can describe a larger variety of semantic mappings between terms, not just equivalence. Similarly, [14] presents a case study in the medical domain where mappings between two ontologies are inferred from manually established mappings with a third ontology, and by using the reasoning mechanisms permitted by the C-OWL language.

The advantage of these approaches is that they use richly axiomatized ontologies as background knowledge and therefore guarantee the semantic nature of the mappings. However, a weakness is that the appropriate reference ontology needs to be manually selected prior to mapping. As already pointed out, in many scenarios this approach is unfeasible as we might not know in advance which terms from which ontologies we may want to map. Even in the cases where a reference ontology can be manually selected prior to performing the mapping, there is no guarantee that such an ontology actually exists.

3. Online textual resources can provide an important source of background knowledge. van Hage et. al [15] rely on the combination of two "linguistic ontology mapping techniques" that exploit online available textual sources to resolve mappings between two thesauri in the food domain. On the one hand, they use Google to determine subclass relationships between pairs of concepts using the Hearst pattern based technique introduced by the PANKOW system [4]. On the other hand, they exploit the regularities of an online cooking dictionary to learn hypernym relations between concepts of the source and target ontologies.

The strength of this approach is that it reduces the high cost of establishing adequate background knowledge. Indeed, the background knowledge sources are dynamically discovered and used [15]. There is no need for a manual and domain dependent ontology selection task prior to mapping. The drawback is that the right knowledge has to be extracted first. However, knowledge extraction techniques generally lead to considerable noise and so, reduce the quality of the mapping (e.g., $Mayonnaise \sqsubseteq Cold$). Therefore, without human validation, online texts cannot be considered as reliable semantic resources.

We conclude that the use of background knowledge overcomes the major limitations of syntactic approaches: it allows obtaining semantic relations even between dissimilar ontologies. However, existing approaches either 1) rely on an a priori selected reference ontology or, if they acquire knowledge dynamically, 2) suffer from the noise introduced by knowledge extraction techniques. As a result, they are not suitable for use by novel Semantic Web tools, such as PowerAqua, which require both that the returned mappings are semantically sound and that the relevant background knowledge is dynamically selected, at run-time. In the next section we describe an approach that fulfills these requirements.

3 Using The Semantic Web as Background Knowledge

Our hypothesis is that the growing amount of online available semantic data which makes up the Semantic Web can be used as a source of background knowledge in ontology mapping in a way that satisfies the requirements identified in the previous section. Indeed, this large-scale, heterogeneous semantic data collection provides formally specified knowledge which is likely to be less faulty than that derived from textual sources and therefore lead to better mappings. Moreover, the size and heterogeneity of the collection makes it possible to dynamically select and combine the appropriate knowledge and to avoid the manual selection of a single, large ontology. In the following we investigate increasingly sophisticated approaches to discover and exploit online available ontologies for mapping. We also provide experimental evidence that such mappings can be obtained.

Experimental Data. We have used the dataset described in [15] for our experiments. In their work, van Hage et al. compare the UN FAO's AGROVOC[2] and the USDA Nutrient Database for Standard Reference, release 16 (SR-16)[3] thesauri. Their mapping techniques are verified on a subset of these thesauri. Two modules are selected from AGROVOC one describing food types (A-Food, 21 concepts), the other describing animal products (A-Animal, 88 concepts). These are compared against one module of SR-16 describing meat products (SR-Meat, 24 concepts). Together with the terms (concept names) of these modules, the authors provided us with manually established alignments. The 32 mappings from A-Food to SR-Meat and the 31 mappings from A-Animal to SR-Meat are used here as gold standards for validating the results of our technique.

Implementation Details. We explore our idea by implementing different mapping strategies on top of the Swoogle'05 ontology search engine [5]. Swoogle crawls and indexes a large amount of semantic metadata available online and as such allows access to a large part of the Semantic Web.

Notations. Each strategy takes two candidate concept names (`A` and `B`) as an input and returns the discovered mapping between them. The corresponding concepts in the selected ontology are `A'` and `B'` ("anchor terms"). We rely on the description logic syntax for semantic relations occurring between concepts in an ontology, e.g., $\mathtt{A'} \sqsubseteq \mathtt{B'}$ means that `A'` is a sub-concept of `B'` in a selected ontology and $\mathtt{A'} \perp \mathtt{B'}$ means that `A'` and `B'` are disjoint. The returned mappings are expressed using C-OWL [2] like notations, like $\mathtt{A} \xrightarrow{\sqsubseteq} \mathtt{B}$ or $\mathtt{A} \xrightarrow{\perp} \mathtt{B}$.

3.1 S1: Mappings Based on One Ontology

Our simplest strategy consists in using Swoogle to find ontologies containing concepts with the same names as the candidate concepts and to derive mappings from their relationship in the selected ontologies. Figure 1(b) illustrates this strategy with an example where three ontologies are discovered containing the concepts `A'` and `B'` with the same names as `A` and `B`. The first ontology contains no relation between the anchor concepts, while the other two ontologies contain a subsumption relation. The concrete steps of this strategy are:

1. Select ontologies containing concepts `A'` and `B'` corresponding to `A` and `B`;
2. For each resulting ontology:
    - if $\mathtt{A'} \equiv \mathtt{B'}$ then derive $\mathtt{A} \xrightarrow{\equiv} \mathtt{B}$;
    - if $\mathtt{A'} \sqsubseteq \mathtt{B'}$ then derive $\mathtt{A} \xrightarrow{\sqsubseteq} \mathtt{B}$;
    - if $\mathtt{A'} \sqsupseteq \mathtt{B'}$ then derive $\mathtt{A} \xrightarrow{\sqsupseteq} \mathtt{B}$;

---

[2] `http://www.fao.org/agrovoc`

[3] `http://www.nal.usda.gov/fnic/foodcomp/Data/SR16/sr16.html`

  - if `A'` $\perp$ `B'` then derive `A` $\xrightarrow{\perp}$ `B`;
3. If no ontology is found, no mapping is derived;

Even if this strategy seams simple, it leads to several implementation choices, depending on the relative importance given to time performance and accuracy of the mapping mechanism:

Stop when the first mapping is found. In its simplest version, the algorithm would stop as soon as a mapping is discovered. This is the easiest way to deal with the multiple returned ontologies but it assumes that the first discovered relation can be trusted and there is no need to inspect the other ontologies. Note that the first ontology returned by Swoogle does not necessarily contain a relation between the candidate concepts (like in the example Figure 1(b)). Here we use the first ontology containing such a relation, but, in another implementation, it could be considered that if an ontology covers the candidate concepts without relating them, then no mapping should be derived.

Dealing with contradictions. Instead of relying on the information provided by only one ontology as before, we can envisage to combine the results obtained using all the selected ontologies. Mappings resulting from different sources can be different (e.g., `A` $\xrightarrow{\sqsubseteq}$ `B` and `A` $\xrightarrow{\sqsupseteq}$ `B`), or, in the worst case, inconsistent (e.g., `A` $\xrightarrow{\sqsubseteq}$ `B` and `A` $\xrightarrow{\perp}$ `B`). Several ways of dealing with these contradictions can be considered: we can keep all the mappings (favoring recall), only keep mappings without contradiction (favoring precision), keep the mappings that are derived from most of the ontologies, or try to combine the results (e.g., by deriving `A` $\xrightarrow{\equiv}$ `B` from `A` $\xrightarrow{\sqsubseteq}$ `B` and `A` $\xrightarrow{\sqsupseteq}$ `B`). In any case, combining the results from several ontologies is more time consuming (but more reliable) than deriving it from a single ontology.

Considering a particular level of inferences. In the simplest implementation, we can rely on *direct* and *declared* relations between `A'` and `B'` in the selected ontology. But, for better results, *indirect* and *inferred* relations should also be exploited (e.g., if `A'` $\sqsubseteq$ `C` and `C` $\perp$ `B'`, then `A'` $\perp$ `B'`). Different levels of inferences can be considered (no inference, basic transitivity, DL reasoning), each of them representing a particular compromise between the performance of mapping and the completeness of the result.

Experimental results. For our experiments, we implemented this first strategy using basic transitivity reasoning (i.e., taking into account all parents of `A'` and `B'`) and stopping as soon as a relation was found.

A-Food vs. SR-Meat: We obtained three mappings for these term sets: $Beef, Pork, Poultry \xrightarrow{\sqsubseteq} Food$. All mappings were derived from the Tap ontology[4], where, for example, $Beef \sqsubseteq ReadMeat \sqsubseteq MeatOrPoultry \sqsubseteq Food$.

A-Animal vs. SR-Meat: For these hierarchies, our implementation yielded in a single mapping, $Bacon \xrightarrow{\sqsubseteq} Pork$, which can be found as is in Tap.

Analyzing our results, we discovered that a key factor in the efficiency of our approach is the level to which the candidate terms are covered by Swoogle.

---

[4] `http://139.91.183.30:9090/RDF/VRP/Examples/tap.rdf`

Indeed, comparing our results to the gold standard mappings, we observed that 24 out of 32 for A-Food vs. SR-Meat (and 20 out of 31 for A-Animal vs. SR-Meat) involve concepts that do not exist in any ontology known to Swoogle (e.g., $GuineaHen, Quail, Squab$). Our experiments are quite strict with respect to finding anchor terms for the candidate concepts: only concepts with identical names are considered. In the next section we suggest ways to reduce this problem.

3.2 S2: Extending Swoogle's Coverage

In order to discover more ontologies that cover the candidate concepts, the process of finding anchor terms must be more flexible. This flexibility can be achieved by considering the following techniques:

A. String normalization. Differences between concept names can be based on simple differences in naming conventions (e.g., $TURKEY\_BREAST$ and $TurkeyBreast$). Most mapping mechanisms use *string normalization* techniques, that consist in transforming strings into a standard form before comparison. Our ontology selection relies on such mechanisms as well.

B. Dealing with compound names. Compound names are particularly difficult to match as they are likely to appear under slightly different forms. Several mapping techniques suggest to be more flexible when searching for compound terms and to allow for:

Different order of the constituents. For example, the term $TurkeyRoast$ does not appear in Swoogle, but *RoastTurkey* does.

Additional constituents. For example, $TurkeyBreast$ is not covered but $TurkeyMeatBreast$ (which additionally contains $Meat$) is.

Less constituents. Some compound terms are only partially covered. For example, *MeatProduct* does not exist in Swoogle, but $Meat$ does.

Such a flexible matching is also used when discovering anchor terms in the work of Aleksovski et al. [1]. However, while the examples given above are semantically equivalent, automatically identifying lexically different but semantically equivalent compound terms is a difficult task.

B. Exploiting semantic relations between terms. Semantic relations such as synonymy can be used to replace terms with their semantic equivalents. A good source for synonymy information is WordNet. However, the drawbacks of WordNet are that it is difficult to get relevant synonyms unless the sense of the term is known a priory and that compound terms are weakly covered.

Experimental results. Just to prove the point that extended coverage can have a significant effect on the obtained mappings, we rerun our experiments by replacing some terms with their syntactic approximates. We replaced *TurkeyRoast* with *RoastTurkey* (SR-Meat), *TurkeyBreast* with $TurkeyMeatBreast$ (SR-Meat) and, *MeatProduct* with $Meat$ (A-Animal).

A-Food vs. SR-Meat: We obtained that *RoastTurkey* $\xrightarrow{\sqsubseteq}$ $PreparedFood$ and *RoastTurkey* $\xrightarrow{\sqsubseteq}$ *Food* because *RoastTurkey* $\sqsubseteq$ *TurkeyDish* $\sqsubseteq$ *PoultryDish* $\sqsubseteq$ *MeatDish* $\sqsubseteq$ *PreparedFood* $\sqsubseteq$ *Food*. Also, $TurkeyMeatBreast \xrightarrow{\sqsubseteq}$ *Food*. All mappings were derived from Tap.

A-Animal vs. SR-Meat: We obtained three extra mappings: $Beef \xrightarrow{\sqsubseteq} Meat$[5], $Ham \xrightarrow{\sqsubseteq} Meat$[6] and $Pork \xrightarrow{\sqsubseteq} Meat$[7].

Increasing the coverage of the mapped terms by replacing them with semantically similar variants leads to more mappings. However, another problem comes from the fact that both candidate concepts might not appear in a single ontology, even if each of them appears by itself in many ontologies. Therefore, another way to obtain more mappings is to extend the matching strategy to combine information derived from multiple ontologies, as detailed next.

3.3 S3: Cross-Ontology Mapping Discovery

The previous strategies (S1 and S2) assume that a semantic relation between the candidate concepts can be discovered in a single ontology. However, some relations could be distributed over several ontologies. Therefore, if no ontology is found that relates both candidate concepts, then the mappings should be derived from two (or more) ontologies. In this strategy, mapping is a recursive task where two concepts can be mapped because the concepts they relate in some ontologies are themselves mapped (Figure 1(c)):

1. If no ontologies are found that contain both `A` and `B` then select all ontologies containing a concept `A'` corresponding to `A`;
2. For each of the resulting ontologies:
   (a) for each `C` such that $\mathtt{A'} \sqsubseteq \mathtt{C}$, search for mappings between `C` and `B`;
   (b) for each `C` such that $\mathtt{A'} \sqsupseteq \mathtt{C}$, search for mappings between `C` and `B`;
   (c) derive mappings using the following rules:
      - (r1) if $\mathtt{A'} \sqsubseteq \mathtt{C}$ and $\mathtt{C} \xrightarrow{\sqsubseteq} \mathtt{B}$ then $\mathtt{A} \xrightarrow{\sqsubseteq} \mathtt{B}$
      - (r2) if $\mathtt{A'} \sqsubseteq \mathtt{C}$ and $\mathtt{C} \xrightarrow{\equiv} \mathtt{B}$ then $\mathtt{A} \xrightarrow{\sqsubseteq} \mathtt{B}$
      - (r3) if $\mathtt{A'} \sqsubseteq \mathtt{C}$ and $\mathtt{C} \xrightarrow{\perp} \mathtt{B}$ then $\mathtt{A} \xrightarrow{\perp} \mathtt{B}$
      - (r4) if $\mathtt{A'} \sqsupseteq \mathtt{C}$ and $\mathtt{C} \xrightarrow{\sqsupseteq} \mathtt{B}$ then $\mathtt{A} \xrightarrow{\sqsupseteq} \mathtt{B}$
      - (r5) if $\mathtt{A'} \sqsupseteq \mathtt{C}$ and $\mathtt{C} \xrightarrow{\equiv} \mathtt{B}$ then $\mathtt{A} \xrightarrow{\sqsupseteq} \mathtt{B}$

In this strategy, steps (a) and (b) can be ran in parallel and stopped when one of them is able to establish a mapping. These two steps correspond to the recursive part of the algorithm. The task of *searching for mappings between `C` and `B`* can be realized using one of our three strategies.

Experimental results. We have implemented this algorithm by using the first mapping strategy (S1) in the recursive part.

A-Food vs. SR-Meat: By combining information available in different ontologies, we obtained that $Chicken, Duck, Goose, Turkey \xrightarrow{\sqsubseteq} Food$ because they are subclasses of *Poultry* in some ontologies [8] and $Poultry \sqsubseteq Food$ in

---

[5] `http://reliant.teknowledge.com/DAML/Mid-level-ontology.daml`

[6] `http://www.pizza-to-go.org/ontology`

[7] `http://reliant.teknowledge.com/DAML/Economy.daml`

[8] e.g., `http://reliant.teknowledge.com/DAML/Mid-level-ontology.daml`

Tap (r1). We also discovered that $Ham \xrightarrow{\sqsubseteq} Food$ because $Ham \sqsubseteq Meat$ and $Meat \sqsubseteq Food$ in SUMO[9] (r1). Finally, we found that $Ham \xrightarrow{\perp} Seefood$ because $Ham \sqsubseteq Meat$ and $Meat \perp Seafood$[10] (r3).
A-Animal vs. SR-Meat: Because $Beef, Ham, Pork \sqsubseteq Meat$ and $Meat \perp Seafood$ we derive that $Beef, Ham, Pork \xrightarrow{\perp} Seafood$ (r3). Note that incompatibility mappings are not specified in the gold standard.

4 Conclusions

The aim of this paper was to show the feasibility and the potential advantages of using automatically selected online ontologies as background knowledge for semantic mapping. As Table 1 shows, our experiments on two real life examples have provided promising results, which are consistent with our idea of semantic mappings, as discussed in Section 2. In particular, the output of our algorithm provides mappings, which i) are expressed in terms of semantic relations (subsumption, disjunction); ii) rely on semantics, as expressed in external ontologies; and iii) in many cases would have not been discovered by syntactic techniques (e.g., because the strings denoting similar concepts are very different).

	Mappings	
	A-Food vs. SR-16	**A-Animal vs. SR-16**
S1	$Beef, Pork, Poultry \xrightarrow{\sqsubseteq} Food$	$Bacon \xrightarrow{\sqsubseteq} Pork$
S2	$RoastTurkey \xrightarrow{\sqsubseteq} Food, PreparedFood$ $TurkeyMeatBreast \xrightarrow{\sqsubseteq} Food$	$Beef, Ham, Pork \xrightarrow{\sqsubseteq} Meat$
S3	$Chicken, Goose, Turkey, Duck \xrightarrow{\sqsubseteq} Food$ $Ham \xrightarrow{\sqsubseteq} Food$; $Ham \xrightarrow{\perp} Seefood$*	$Beef, Ham, Pork \xrightarrow{\perp} Seafood$*
Total	11 (+1*)	4 (+3*)

Table 1. Discovered mappings. Marked mappings* do not exist in the Gold Standard.

Note that the technique presented here is not meant to be used in isolation, as an alternative to current approaches. On the contrary, we plan to integrate our technique with "syntactic" techniques, to develop a robust and comprehensive ontology mapping method. For this reason it is difficult to provide a detailed comparison with other approaches, using standard measures of precision and recall. Because our technique is meant to enhance, rather than replace existing methods, it scores very highly on precision but relatively low on recall, about 30% on the test case provided by [15] (low recall is due to the fact that many concepts are not covered by any online ontology). Having said so, we should also emphasize that some of the mappings our system is able to discover are not even covered by the Gold Standard defined in [15], which is an indication of the greater range of mapping possibilities provided by our approach.

---

[9] `http://reliant.teknowledge.com/DAML/SUMO.owl`
[10] `http://ontolingua.stanford.edu/doc/chimaera/ontologies/wines.daml`

The broader context of our work is one of providing meaningful mappings that can be used by the next generation of Semantic Web applications [10] to reason over multiple ontologies. Hence, in contrast with most existing work on ontology mapping, we are interested in developing an approach which can be used by systems that need to create mappings dynamically and in real time to make use of the large scale semantics available on the Web. This aspect still needs to be evaluated using appropriate experimental settings and criteria.

Another goal of this paper was to identify some of the research issues brought up by the innovative aspects of our technique. The first innovative aspect is that appropriate background knowledge is automatically selected from the variety of ontologies available on the Semantic Web. As a result, important issues for us concern i) the current level of semantic coverage of the Semantic Web and ii) the quality of the tools that give access to it. Regarding the second point, although Swoogle [5] is by far the most advanced ontology search engine available today, it is still rather limited with respect to supporting our needs to exploit online ontologies dynamically and in real time. Among other things, we need better query facilities, a richer set of relations between ontologies (at the very least to quickly discard duplicate ontologies), and other ranking mechanisms in addition to popularity. Regarding the current level of coverage on the Semantic Web, our previous work [12] indicated a *knowledge sparseness phenomenon*: some domains are well covered by existing ontologies (e.g., academic research and medicine), while others are not covered at all. This phenomenon has a direct influence on our method as coverage of a domain is a prerequisite for successfully mapping ontologies in this domain. Having said so, there is evidence that the Semantic Web is rapidly growing and, as a result, our method will be able to perform better and better, simply by taking advantage of the improved semantic coverage.

The second innovative aspect of our technique, that of combining facts from different ontologies, leads to another important issue: how to deal with contradictions. Online ontologies are made for different purposes, in different contexts and therefore, can lead to contradictory (or inconsistent) mappings. As already mentioned, one of the advantages of semantic techniques is that resulting mappings can be justified and explained. In that sense, one way to deal with contradictions would be to relate mappings to the ontologies on which they are based. Using this solution, contradictory mappings would still co-exist but in a *contextualized* form, i.e., justified and valid only in the context of some particular ontologies. Another way to deal with contradictions would be to rely only on ontologies sharing a similar context with the mapped ones. Indeed, when trying to map the $Turkey$ concept, we would more likely find relevant mappings using ontologies also containing concepts like $Food$ or $Meat$, than ontologies covering countries. This implies that more advanced ontology selection techniques are needed which can consider similarity between ontologies as a selection criterion.

In addition to tackling the aforementioned issues, the experiments presented here need to be followed by several studies. A key next step concerns the complete implementation and evaluation of our technique, which is currently restricted to subsumption and disjunction relations between concepts. Our plan is to extend the technique to also map properties and individuals, so to increase the range of discovered mappings. Finally, when presenting our strategies we emphasized the trade-off between the performance and the accuracy of the mapping mechanism. Finding a good compromise between these two aspects is a hard task and we plan to address this issue by reformulating our technique as an *anytime* algorithm.

Acknowledgements

We are grateful to Willem van Hage for providing us with his experimental data set. This work was funded by the AKT project sponsored under grant number GR/N15764/01, and the Open Knowledge and NeOn projects sponsored under EC grant numbers IST-FF6-027253 and IST-FF6-027595.

References

1. Z. Aleksovski, M. Klein, W. ten Katen, and F. van Harmelen. Matching Unstructured Vocabularies using a Background Ontology. In S. Staab and V. Svatek, editors, *Proc. of EKAW*, LNAI. Springer-Verlag, 2006.
2. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. Contextualizing ontologies. *Journal of Web Semantics*, 1(4):24, 2004.
3. P. Bouquet, L. Serafini, and S. Zanobini. Peer-to-Peer Semantic Coordination. *Journal of Web Semantics*, 2(1), 2005.
4. P. Cimiano, S. Handschuh, and S. Staab. Towards the Self-Annotating Web. In *Proc. of the 13th International World Wide Web Conference*, 2004.
5. L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and Ranking Knowledge on the Semantic Web. In *Proc. of ISWC*, 2005.
6. J. Euzenat, H. Stuckenschmidt, and M. Yatskevich. Introduction to the Ontology Alignment Evaluation 2005. In *Proc. of the Integrating Ontologie WS*, 2005.
7. F. Giunchiglia, P. Shvaiko, and M. Yatskevich:. Semantic Schema Matching. In *Proc. of CoopIS'05*, volume 3760 of *LNCS*, pages 347 – 360, 2005.
8. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Discovering Missing Background Knowledge in Ontology Matching. In *Proc. of ECAI*, 2006.
9. V. Lopez, E. Motta, and V. Uren. PowerAqua: Fishing the Semantic Web. In *Proc. of ESWC*, 2006.
10. E. Motta and M. Sabou. Next Generation Semantic Web Applications. In *Proc. of the 1st Asian Semantic Web Conference (ASWC)*, 2006.
11. N. Noy. Semantic Integration: A Survey of Ontology-based Approaches. *SIGMOD Record*, 33(4), 2004.
12. M. Sabou, V. Lopez, and E. Motta. Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner? In *Proc. of EKAW*, 2006.
13. P. Shvaiko and J. Euzenat. A Survey of Schema-based Matching Approaches. *Journal on Data Semantics*, IV, 2005.
14. H. Stuckenschmidt, F. van Harmelen, L. Serafini, P. Bouquet, and F. Giunchiglia. Using C-OWL for the Alignment and Merging of Medical Ontologies. In *Proc. of the First Int. WS. on Formal Biomedical K. R. (KRMed)*, 2004.
15. W. van Hage, S. Katrenko, and G. Schreiber. A Method to Combine Linguistic Ontology-Mapping Techniques. In *Proc. of ISWC*, 2005.

Exploiting the structure of background knowledge used in ontology matching

Zharko Aleksovski[1,2], Warner ten Kate[1], and Frank van Harmelen[2]

[1] Philips Research, Eindhoven,
[2] Vrije Universiteit, Amsterdam

Abstract. We investigate the use of a background knowledge ontology in ontology matching. We conducted experiments on matching two medical ontologies using a third extensive one as background knowledge, and compare the results with directly matching the two ontologies. Our results indicate that using background knowledge, in particular the exploitation of its structure, has enormous benefits on the matching. The structure of the background ontology needs closer examination to determine how to use it in order to obtain maximal benefit.

1 Introduction

The problem of ontology alignment (also known as ontology integration, semantic integration, ontology mapping, etc.) plays a central role in the development of knowledge based systems. New technologies such as Semantic Web make it easier to use ontologies in the information systems. These trends have driven the development of new ontologies, which in turn has resulted in an increasing amount of ontologies becoming available in the recent years. Essential to an ontology is its reusability, which implies one needs to integrate it into the system using it. Problem arises if the ontology to be integrated uses a different vocabulary from the system using it.

This problem initiated a lot of research on ontology matching lately, see [1–3]. Various approaches have been developed. They mainly focus on two aspects: lexically matching the elements of the ontologies, and using the structure of the ontologies. The first uses string-based and linguistic methods to detect relatedness between elements based on string similarity of their labels, and the second uses the relations within the ontologies to detect similarities. Elements in the ontologies that are related but have neither lexical nor structural similarity remain undetected. Motivated by this issue, we focused on using background knowledge. We followed the intuition that a background ontology which comprehensively describes the domain of the source and target ontologies will provide a way to find matches missed by other approaches.

In earlier papers, we showed that the use of background ontology can compensate for lack of structure and lexical overlap, and increasing the amount of background knowledge (multiple ontologies) improves the matching result, see [4, 5]. In this work we investigate the benefits and problems of using a *comprehensive* domain ontology as background knowledge. We conducted experiments of matching one medical ontology to another, while using a much larger and detailed ontology of the same domain as background knowledge. The results of our experiments confirmed that the background knowledge can significantly boost the performance of the matching process. In particular, maximal benefit is achieved when combining different pieces of knowledge within the background knowledge. However, these pieces need careful consideration when combining them.

The paper is organized as follows: in Section 2 we describe the general scheme of our approach, that is how we use an ontology as background knowledge in ontology integration. In Sections 3 and 4 we describe a case study and a set of experiments to test our expectations. In Section 5 we report on the results of these experiments with evaluation on validity. In Section 6 we discuss representative matches of the different experiments, and Sections 7 and 8 conclude the paper with discussion on related and future work.

2 Our approach: Using a background knowledge ontology in ontology matching

In our approach we match two ontologies using a third as background knowledge. We call the ontologies being matched the source and target, see Figure 1. We make use of the background knowledge by first relating the concepts from the source and target ontology to the background knowledge, and then checking if these concepts are related. Hence, this process proceeds in two steps: anchoring and deriving relations.

Anchoring is matching the source and target concepts to the background knowledge. In general, this process can be performed by using an existing ontology matching technique. Besides the concept's labels one can also use the structure of the ontologies. In the anchoring we are not only interested in finding the corresponding equivalent concepts. As we will see in our experiments later, other kind of relatedness with the concepts in the background knowledge can be useful as well.

Deriving relations is the process of discovering relations between source and target concepts by looking for relations between their anchored concepts in the background knowledge. Both the source and target concept's anchors are part of the background knowledge, and checking if they are related means using the reasoning service in the background knowledge ontology. Combining the anchor relations with the relations between the background knowledge concepts derives the relation between source and target concepts, which is what we are looking for.

To explain this process in the context of medical ontologies, a realistic example is the following: the source concept $\mathcal{SRC}$*:Brain* is anchored to background knowledge concept $\mathcal{BK}$*:Brain*, and the target concept $\mathcal{TAR}$*:Head* is anchored to a background knowledge concept $\mathcal{BK}$*:Head*. The background knowledge reveals a relation $\mathcal{BK}$*:Brain part-of* $\mathcal{BK}$*:Head*, and we derive a relation that source concept $\mathcal{SRC}$*:Brain* has a narrower meaning than the target concept $\mathcal{TAR}$*:Head*. Using background knowledge was crucial in this case; the match was not found by directly matching the source to the target ontology, $\mathcal{SRC}$*:Brain* is classified under $\mathcal{SRC}$*:Central nervous system* which is in no way related to the concept $\mathcal{TAR}$*:Head*.

As the example suggests, of particular interest in our approach is exploiting the structure of the background knowledge ontology. It is done in the deriving relations step, when checking for relatedness between the anchored concepts in the background

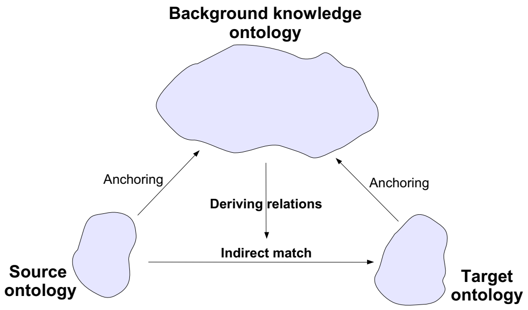


Fig. 1. Matching source to target ontology using background knowledge.

knowledge ontology. Now we introduce the formal definitions of all the components in this scheme, which we will use in the rest of the paper.

Concept is a class of things grouped together due to some shared property. It is named with one or more labels which are synonymous to each other. Besides the labels, concept is also determined by its relations to other concepts. We will refer to a concept in two ways: with capital italic letters X, Y... when referring to an arbitrary concept, $X^{\mathcal{O}}$ or $X^{\mathcal{ONT}}$ when referring to a concept from a specific ontology, and by using its label, for example *Temporal lobe*, and $\mathcal{ONT}$:*Temporal lobe* or $\mathcal{O}$:*Temporal lobe* when referring to a concept from a specific ontology.

Relation is a triple $(X, \mathit{relation}, Y)$, where X and Y are concepts, and $\mathit{relation} \in T$, where T is the set of all types of relations. We will also write it as $X \sim Y$ with "$\sim$" being the symbol of the relation. Examples of relations used in this paper are: $X \equiv Y$, - the two concepts have the same meaning, and $X \preceq Y$ (with inverse: $\succeq$), also written as X *is-narrower-than* Y - the first has narrower meaning than the second. Other relations are used in the existing ontologies as well, see Section 4. Relations can be established between concepts from the same and also from different ontologies.

Ontology is a pair of sets: $\mathcal{ONT}(C, R)$. C is a set of concepts, R is the set of relations among these concepts. We will refer to an ontology using shortened form of its name written in calligraphic letters, like $\mathcal{ONT}$. When referring to C or R of a specific ontology, we will write them as $C^{\mathcal{ONT}}$ and $R^{\mathcal{ONT}}$

Ontology match is a function of two ontologies that returns a set of relations between their concepts:

$$f : (\mathcal{SRC}, \mathcal{TAR}) \rightarrow \{(X, \mathit{relation}, Y) | X \in C^{\mathcal{SRC}}, \mathit{relation} \in T, Y \in C^{\mathcal{TAR}}\} \quad (1)$$

Specific types of ontology matches of interest to our approach are the following two: **Anchorings** are two ontology matches from the source and target ontology to the background knowledge respectively, and **Deriving relations** is an ontology match between the source and the target ontology which is an indirect matching that uses their anchors to the background knowledge, and the background knowledge itself.

3 Our case study

Hypothesis: Using comprehensive domain ontology as background knowledge can significantly boost the performance of an ontology matching process.

To test this hypothesis and investigate the problems that occur when our matching scheme is used in practice, we conducted a set of experiments matching existing ontologies available on the Semantic Web. We matched the anatomy parts of CRISP and MeSH using the FMA ontology as a background knowledge. CRISP and MeSH were choosen randomly, and FMA because it extensively covers the anatomy domain.

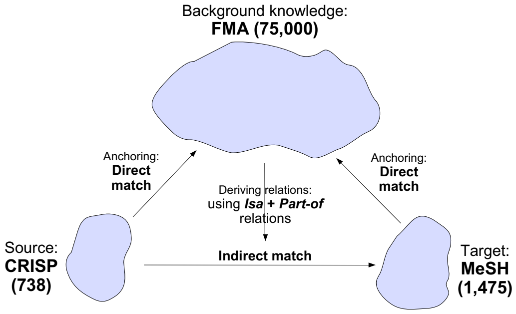


Fig. 2. Matching CRISP to MeSH ontology using FMA as background knowledge.

The test data used in the experiments.

Source ontology: **CRISP**[3] (Computer Retrieval of Information on Scientific Projects) is a biomedical document classification system. It contains 738 concepts organized in a hierarchy. The relations in its hierarchy are established based on the classified document sets. The hierarchy contains two relations: *broader-than* and its inverse *narrower-than*, meaning superset and subset between the corresponding document sets respectively. In our experiments we used the part of CRISP describing anatomy.

Target ontology: **MeSH**[4] is the National Library of Medicine's controlled vocabulary thesaurus intended for classification of documents. The part which we used in the experiments is the anatomy sub hierarchy. It contains is 1475 concepts, and is based on *broader-than* and its inverse *narrower-than* relations, the same as CRISP.

Background knowledge ontology: **FMA**, as stated in its description[5]: "The Foundational Model of Anatomy is a domain ontology that represents a coherent body of explicit declarative knowledge about human anatomy." The version of FMA used in our experiments dates from the end of 2005. It contains 75000 concepts interconnected with around 160 different relation types. We used the main two hierarchies: *isa* and *part-of*.

---

[3] http://crisp.cit.nih.gov/

[4] http://www.nlm.nih.gov/mesh/

[5] http://sig.biostr.washington.edu/projects/fm/AboutFM.html

Testing our hypothesis by conducting experiments.
We performed five experiments matching CRISP to MeSH. In the first we matched them directly, and in the other four we matched them indirectly using the FMA ontology as background knowledge. The direct matching served as a baseline, against which we compared each of the indirect matchings. With the intention to distile the added value of using background knowledge, we analyzed the additional matches discovered by indirect matching. Furthermore, there were some cases of matches found in the direct and not in the indirect matching. We discuss these in Section 6.

4 The experiments

We performed five experiments. In Experiment 1 we matched CRISP to MeSH directly and in the other four we matched them indirectly using FMA as background knowledge. In Experiment 2 we exploited each *isa* and *part-of* relation in FMA separately; in Experiment 3 we used *isa* and *part-of* relations with their transitive closures; in Experiment 4 we used *isa* and *part-of* combined, and in Experiment 5 we restricted to specific combinations of them to induce the matches. The result of each experiment was an ontology match between CRISP and MeSH concepts, using one of the three relations: $\equiv$ (*equivalent*), $\preceq$ (*narrower-than*), $\succeq$ (*broader-than*). Now, we explain the direct and then the indirect matchings in detail.

Experiment 1: Direct matching was performed in two steps: lexical and structural. In the lexical phase we matched CRISP to MeSH using the concept's labels. We cleaned the labels of interpunction, general words like *the*, *of*, *and* etc., we accounted for word order and singular/plural forms of the same words. When matching two concepts X and Y, we concluded $X \equiv^d Y$ if a pair of their labels matches[6]. Further, we used partial matches as well, if X has a label consisting of a superset of words of a label of Y we concluded $X \preceq^d Y$, and analogously $X \succeq^d Y$ if X has label of subset of the words of a label of Y. In other words, we used the partial lexical matches following the intuition that additional words in a label additionally constrain the meaning of that concept. This way, for example, we concluded that $\mathcal{CSP}$*: Mesenteric artery* $\preceq^d$ $\mathcal{MSH}$*: Artery*.

In the structural phase of direct matching we used the structure of CRISP and MeSH to further induce matches by combining the relations from CRISP and MeSH and the lexical matches. For example, from the two relations:

- $\mathcal{CSP}$: *Brain* $\equiv^d$ $\mathcal{MSH}$: *Brain*
- $\mathcal{MSH}$: *Brain* $\succeq$ $\mathcal{MSH}$: *Temporal lobes*

we can induce the relation

- $\mathcal{CSP}$: *Brain* $\succeq^d$ $\mathcal{MSH}$: *Temporal lobes*

We extended the set of lexical matches with the matches implied by the structure of CRISP and MeSH. The following rules were used to extend the result set[7]:

---

[6] The small d letter in the right upper corner means that the relation is a direct match, letter a means it is an anchor relation, and letter i means that it is an indirect match.

[7] $X^{\mathcal{C}}$, $X^{\mathcal{M}}$, $X^{\mathcal{F}}$ stand for an arbitrary concept from CRISP, MeSH and FMA respectively

- if $(X^{\mathcal{C}} \preceq^d Y^{\mathcal{M}}) \wedge (Y^{\mathcal{M}} \preceq Z^{\mathcal{M}})$ induce $(X^{\mathcal{C}} \preceq^d Z^{\mathcal{M}})$
- if $(X^{\mathcal{C}} \preceq Y^{\mathcal{C}}) \wedge (Y^{\mathcal{C}} \preceq^d Z^{\mathcal{M}})$ induce $(X^{\mathcal{C}} \preceq^d Z^{\mathcal{M}})$
- if $(X^{\mathcal{C}} \succeq^d Y^{\mathcal{M}}) \wedge (Y^{\mathcal{M}} \succeq Z^{\mathcal{M}})$ induce $(X^{\mathcal{C}} \succeq^d Z^{\mathcal{M}})$
- if $(X^{\mathcal{C}} \succeq Y^{\mathcal{C}}) \wedge (Y^{\mathcal{C}} \succeq^d Z^{\mathcal{M}})$ induce $(X^{\mathcal{C}} \succeq^d Z^{\mathcal{M}})$

These rules also used $\equiv$ relations, where $X \equiv Y$ was considered as $X \preceq Y$ and $X \succeq Y$. The rules were exhaustively applied on the result set.

Indirect matching followed the scheme that we described in Section 2. It was performed in two steps: first anchoring CRISP and MeSH to FMA, and then deriving relations between CRISP and MeSH using FMA as background knowledge, Figure 2.

In the anchoring we used the same direct matching technique described for matching CRISP to MeSH directly. Both CRISP and MeSH were anchored to FMA. The result was set of matches with three different kinds of relations: $X \equiv^a Y$, $X \preceq^a Y$, $X \succeq^a Y$, where Y is a concept from FMA, and X is in CRISP or MeSH.

When deriving the relations we used the following rules:

- if $(X^{\mathcal{C}} \preceq^a Y^{\mathcal{F}}) \wedge (Y^{\mathcal{F}} \preceq Z^{\mathcal{F}}) \wedge (Z^{\mathcal{F}} \preceq^a Q^{\mathcal{M}})$ induce $(X^{\mathcal{C}} \preceq^i Q^{\mathcal{M}})$
- if $(X^{\mathcal{C}} \succeq^a Y^{\mathcal{F}}) \wedge (Y^{\mathcal{F}} \succeq Z^{\mathcal{F}}) \wedge (Z^{\mathcal{F}} \succeq^a Q^{\mathcal{M}})$ induce $(X^{\mathcal{C}} \succeq^i Q^{\mathcal{M}})$

where we used the relations *isa* and *part-of* for $\succeq$, and their inverse *has-kind* and *has-part* for $\preceq$. However, in FMA there are no *broader-than* and *narrower-than* relations, but their specializations: *isa* and *part-of* with their inverses *has-kind* and *has-part*. We conducted four different experiments of indirect matching while using FMA as a background knowledge. The experiments differ in the way *isa* and *part-of* relations from FMA were used and combined when deriving *broader-than* and *narrower-than* relations which are then used in the two rules stated above to derive the indirect matches between CRISP and MeSH concepts.

Experiment 2: Indirect matching by using FMA *isa* and *part-of* relations without transitive closure. We induced a relation between the FMA concepts if they were directly related with *isa* or *part-of* relation. We used the following rules:

- $(X^{\mathcal{F}}\, \mathit{isa}\, Y^{\mathcal{F}})$ induce $(X^{\mathcal{F}} \preceq Y^{\mathcal{F}})$
- $(X^{\mathcal{F}}\, \mathit{part\text{-}of}\, Y^{\mathcal{F}})$ induce $(X^{\mathcal{F}} \preceq Y^{\mathcal{F}})$

When a relation $X^{\mathcal{F}} \preceq Y^{\mathcal{F}}$ was induced, we added its semantic equivalent $Y^{\mathcal{F}} \succeq X^{\mathcal{F}}$ as well. We did this in all the indirect matching experiments.

Experiment 3: Indirect matching by using FMA *isa* and *part-of* relations with their transitive closures. Relation between two FMA concepts was induced when they were related with the transitive closure of *isa* or *part-of* relations. We used the following rules:

- $(X_1^{\mathcal{F}}\, \mathit{isa}\, X_2^{\mathcal{F}}\, \mathit{isa} \ldots \mathit{isa}\, X_n^{\mathcal{F}})$ induce $(X_1^{\mathcal{F}} \preceq X_n^{\mathcal{F}})$
- $(X_1^{\mathcal{F}}\, \mathit{part\text{-}of}\, X_2^{\mathcal{F}}\, \mathit{part\text{-}of} \ldots \mathit{part\text{-}of}\, X_n^{\mathcal{F}})$ induce $(X_1^{\mathcal{F}} \preceq X_n^{\mathcal{F}})$

Experiment 4: Indirect matching by using the transitive closure of FMA *isa* and *part-of* relations combined. In this experiment we completely merged *isa* and *part-of* relations and then used the transitive closure of the resulting relation. We used one single inference rule:

- $(X_1^{\mathcal{F}}\ \mathit{rel}^1\ X_2^{\mathcal{F}}\ \mathit{rel}^2 \dots \mathit{rel}^{n-1}\ X_n^{\mathcal{F}})$ where $\mathit{rel}^i \in \{\mathit{isa}, \mathit{part\text{-}of}\}$ induce $(X_1^{\mathcal{F}} \preceq X_n^{\mathcal{F}})$

After analyzing the results of Experiment 4 it appeared that false positive matches were created due to using *isa* relation before *part-of* in the process of inducing matches, see Section 6 for clarification. To overcome this negative effect we conducted the next experiment.

Experiment 5: Indirect matching by using the transitive closure of FMA *isa* and *part-of* relations without using *isa* before *part-of*. What we did in this experiment was avoiding the use of *isa* relation before *part-of*. We used one single inference rule:

- $(X_1^{\mathcal{F}}\ \mathit{part\text{-}of}\ X_2^{\mathcal{F}} \dots X_{k-1}^{\mathcal{F}}\ \mathit{part\text{-}of}\ X_k^{\mathcal{F}}\ \mathit{isa}\ X_{k+1}^{\mathcal{F}} \dots X_{n-1}^{\mathcal{F}}\ \mathit{isa}\ X_n^{\mathcal{F}})$ induce $(X_1^{\mathcal{F}} \preceq X_n^{\mathcal{F}})$

5 Results and evaluation

We present now the results of the experiments. First we explain the numbers presented in the tables, then we interpret and explain their meaning, and finally we provide evaluation on the results.

An important issue in presenting the matching results is that in one set of matches many of them may be implied by the others, in combination with the structure of the ontologies. For example, all the concepts in CRISP are found more specific than the root concept in MeSH, whereas having equivalence between the two root concepts already implies all those matches. Similarly, having a match between two concepts contains implicit knowledge about their sub and super-concepts. To make a fair trade-off between the two cases of having all the possible matches and having only the minimal set of matches that implies all the rest, we decided on a result set that is in between.

In each matching experiment we did the following: We started from the set of all matches, including the implied. For each source concept we took the set of all its matches, and then minimized that set by discarding the matches which are implied by the rest of the set. The minimal set is not sensitive to the order of discarding the implied matches. The union of these minimized sets was the final result. This trade-off matching set extracts the minimal knowledge from the matching result for each of the source concepts separately.

5.1 Results of direct and indirect matchings

In the anchoring phase we matched CRISP and MeSH to FMA directly. The results are shown in Figure 3. The equivalence relations were established as 1-1 matches, while narrower-than and broader-than as many to many. Looking for equivalences only already produced successful anchoring: 65.5% of CRISP and 70.6% of MeSH concepts were anchored to their equivalent concepts in FMA. This success comes from the richness of FMA. On the other hand, for many there were no equivalent concepts in FMA because of disagreement on the coverage of anatomy domain. In CRISP there is a concept $\mathcal{CSP}$*: Muscle movement* which is not an anatomical part of the human body, and as such does not exist in FMA. Still, as shown in the last column on Figure 3, nearly 99% of the concepts from both CRISP and MeSH were anchored due to using the structure

	Anchoring concepts	$\equiv$	$\preceq$	$\succeq$	Anchored concepts
Anchoring CRISP to FMA	738	483 (65.5%)	607	1,474	730 (98.9%)
Anchoring MeSH to FMA	1,475	1,042 (70.6%)	1,545	2,227	1,462 (99.1%)

Fig. 3. Anchoring CRISP and MeSH to FMA

of CRISP and MeSH. For example, $\mathcal{CSP}$*: Muscle movement* was anchored as *narrower-than* $\mathcal{FMA}$*: Muscle* because within CRISP it is *narrower-than* $\mathcal{CSP}$*: Muscle*.

Figure 4 summarizes the results of the five experiments. Comparing the indirect to the direct matching, the indirect matchings found many more *narrower-than* and *broader-than* relations than the direct matching. It appeared that the concepts in CRISP and MeSH can be related in many more ways which can not be found by using only the structure of these ontologies alone. In our case FMA contributed the missing knowledge which resulted in such an improvement over the direct matching.

Matches of CRISP to MeSH	$\equiv$	$\preceq$	$\succeq$	$\equiv + \preceq + \succeq$	increase
Exper. 1: Direct	448	417	156	1,021	
Exper. 2: Indir. *isa* and *part-of*	395	516	405	1,316	29%
Exper. 3: Indir. *isa* and *part-of* closure	395	933	1,402	2,730	167%
Exper. 4: Indir. *isa* and *part-of* mixed and closure	395	1,511	2,228	4,143	306%
Exper. 5: Indir. *isa* and *part-of* isa only after part-of	395	972	1,800	3,167	210%

Fig. 4. Matching CRISP to MeSH directly and indirectly

The last column in Figure 4 shows the increase of amount of matches of the indirect matching when compared to the direct matching. The indirect matching of Experiment 2 produced 29% more matches than the direct matching. So, using only the direct *isa* and *part-of* relations between the concepts in FMA already outperformed the direct matching. When using the transitive closure of *isa* and *part-of* (Experiment 3) we obtained increase of 167%, or nearly 2.7 times more matches than the direct matching. When arbitrarialy mixing *isa* and *part-of* with their transitive closure we got increase of 306%, or 4 times more matches than the direct. The fifth experiment, when combining the *isa* and *part-of* in a resticted way, there was an increase of 210% which is 3.1 times more matches than the direct. It produced 26% less matches than the fourth, and 19% more than the third experiment.

These numbers show that using background knowledge produces enormously more matches than direct matching. Without combining the relations within the background knowledge it is already better than the direct matching, then combining the relations in the background knowledge produces much more matches, and combining different relations within background knowledge produces the maximal number. Of course, these numbers do not say anything about the quality of these matches. This will be discussed in the next section. In particular, if the relations are combined arbitrarily then there is big increase in the amount of matches but also false positive matches are created, but when combining them in a specific way we retain the precision of the matches while again considerably increasing the recall.

When looking at *Equivalent* ($\equiv$), the indirect matching found slightly less relations. All the indirect matchings discovered the same amount of equivalences because the only way to find equivalence indirectly is to have both concepts anchored to the same concept in background knowledge. The equivalences found directly and not indirectly were caused by concepts which existed in CRISP and MeSH but not in FMA. In the next section we take a closer look at such a case. In few cases equivalences were detected indirectly and not directly because their labels were found as synonymous only through the background knowledge.

5.2 Evaluation of results

To test for correctness of the matches that we produced with the different experiments, we randomly choose 30 CRISP concepts, and inspected their matches by manually browsing the Wikipedia[8] pages describing these concepts. The evaluation is presented in Figure 5.

	$\equiv$	$\preceq$	$\succeq$	Total	Correct(%)
Exper. 1: Direct	17	18	3	38	38 (100%)
Exper. 4: Indir. *isa* and *part-of* mixed and closure	14	39	59	112	105 (93.7%)
Exper. 5: Indir. *isa* and *part-of* not isa after part-of	14	37	50	101	101 (100%)

Fig. 5. Evaluation of the matchings CRISP to MeSH directly and indirectly on 30 random CRISP concepts

In the last column of Figure 5 is shown the correctness of the matches produced by the different experiments. Only in Experiment 4 there were wrong matches found where the precision dropped to 93.7%, and in the other experiments it was 100% meaning that all the discovered matches were correct. The evaluation of Experiment 2 and 3 was left out because they produced subsets of Experiment 5 which already produced 100% correct matches.

Clearly, 30 concepts is not a sufficient number to get to a full evaluation. However, all these cases were closely examined and gave us the impression that this evaluation depicts the correctness of the experiments. The number of matches on these 30 concepts resembles the ratio as found on the whole test set.

In lack of gold standard, the evaluation phase turned out not to be straight forward. We had to make a choice what to consider correct and what not. Namely, some matches are arguably correct because of the nature of the relation *constitutional-part-of*. For example, the ulnar artery is constitutional part of the elbow, but it also stretches through the whole arm, and therefore it is not part of the elbow only. We call these matches shared.

---

[8] http://wikipedia.org/

Yet, having the relation between ulnar artery and elbow is a useful one, somebody looking for medical resources about an elbow is interested in the arteries passing through the elbow as well, see Wikipedia for more details on this example. We explored the matches $\preceq$ or $\succeq$ produced in Experiment 5, and found out that 30 matches are shared, while the other 57 are not. This means that even if inspecting the matches rigidly by discarding shared matches, the background knowledge still produces a large gain in the matching results.

6 Analysis of discovered matches

We selected three representative cases of matches from the result sets. We will discuss matches found by the indirect matchings and not by the direct, then take a look at the causes for finding incorrect matches when arbitrary mixing the *isa* and *part-of* relations in the background knowledge, and finally we discuss matches found by the direct but not by the indirect matching.

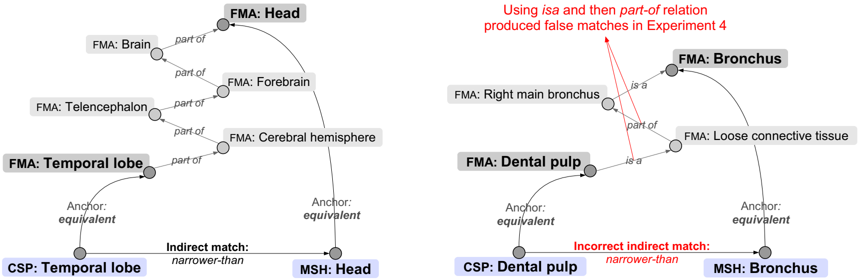


Fig. 6. Two indirect matches: on the left the correct match *Temporal lobe* $\preceq^i$ *Head*, and on the right incorrect match *Dental pulp* $\preceq^i$ *Bronchus*.

Case 1: Matches found by indirect and not by the direct matching. $\mathcal{CSP}$*: Temporal lobe* $\preceq^i$ $\mathcal{MSH}$*: Head* is a representative case of these matches, and is shown on the left hand side in Figure 6. Temporal lobes are parts of the brain, and consequently parts of the human's head. In the structure of MeSH and CRISP they are classified under the Brain which is classified under Central Nervous System, and are not connected in any way with the Head. Therefore relation to the Head is impossible to establish using direct matching, and the background knowledge is crucial in discovering the match.

Case 2: Incorrect match produced by arbitrary mixing of *isa* and *part-of* hierarchies. When using *isa* and then *part-of* in the inference, some of the matches were incorrect. An example is shown on the right hand side in Figure 6 finding $\mathcal{CSP}$:*Dental pulp* $\preceq^i$ $\mathcal{MSH}$:*Bronchus*. Each of the two relations *Dental pulp isa Loose connective tissue part-of Right main bronchus* remain correct when generalizing to *Dental pulp* $\preceq$ *Loose connective tissue*, and *Loose connective tissue* $\preceq$ *Right main bronchus*, but their transitive closure does not hold any more: *Dental pulp* $\preceq$ *Right main bronchus* is incorrect, which then implies the incorrect match $\mathcal{CSP}$:*Dental pulp* $\preceq^i$ $\mathcal{MSH}$:*Bronchus* is found.

Case 3: Matches found by direct but not by indirect matching. An example of such a match is $\mathcal{CSP}$:*Mesenteric artery* $\equiv^d$ $\mathcal{MSH}$:*Mesenteric Arteries*. The relation was not found indirectly because a concept *Mesenteric artery* does not exist in FMA, instead there are two more specific $\mathcal{FMA}$:*Inferior Mesenteric Artery* and $\mathcal{FMA}$:*Superior Mesenteric Artery* and one broader concept $\mathcal{FMA}$:*Artery*. Using FMA the following indirect relations were discovered for $\mathcal{CSP}$:*Mesenteric artery*:

- $\mathcal{CSP}$:*Mesenteric artery* $\preceq^i$ $\mathcal{MSH}$:*Arteries*
- $\mathcal{CSP}$:*Mesenteric artery* $\succeq^i$ $\mathcal{MSH}$:*Mesenteric Artery, Inferior*
- $\mathcal{CSP}$:*Mesenteric artery* $\succeq^i$ $\mathcal{MSH}$:*Mesenteric Artery, Superior*

Combining the results of the direct and indirect matching will improve the result of the both. We showed that in an earlier study in [4].

7 Related work

The research topic of semantic integration is a very active one, yet we encountered two major difficulties when comparing our approach with the others. First is the objective in matching: some target at finding pairs of most corresponding concepts, others aim at 1-1 mappings only, etc.; and second is the different way we use background knowledge.

Existing approaches mostly use background knowledge in the form of lexicons for discovering synonyms, see [6, 2, 7]. S-Match, [8], is example where background knowledge is exploited in doing the mapping, which is very similar to the approach we used in this work. In the current state S-Match uses a predefined set of background knowledge sources, such as Wordnet and UMLS. Moreover, it uses the class hierarchy of the background knowledge ontologies.

Related case-study was performed in [9], where the authors investigated enriching user search queries for image retrieval, by using relations from Wordnet. This follows closely our own scheme: the user search query is a source ontology of one single concept with one single label, Wordnet is the background knowledge, and the classification of images is the target ontology. They used *isa* and *part-of* relations, and they arrived to conclusion comparable with ours about using and combining these relations.

The work we present here is a sequel of another case study that followed similar scheme of using background knowledge in ontology integration, see [4]. In contrast to this work, in the previous studies we used source and target ontologies without structure, the relations we were chasing for were semantic closeness rather than relations with exact semantics such as *broader-than* and *narrower-than*. The background knowledge contained only *broader-than* and *narrower-than* relations, which prohibited us in investigating the impact of combining different relations.

8 Conclusions

Based on the results produced in the experiments, we draw the following conclusions from our study:

(i) Using comprehensive background knowledge in form of ontology can boost the ontology matching process as compared to a direct matching of the two ontologies.

(ii) Most of the value in using background knowledge comes from combining different pieces of knowledge within the background knowledge.

(iii) Different pieces of knowledge within the background knowledge need careful combination in order to gain maximal benefit.

A crucial requirement in using background knowledge is the existence of extensive reference ontologies in different domains at hand. Therefore, the development of such ontologies and subsequent publication on the Semantic Web will make the problem of integration easier.

Currently we are expanding the reported experiments further, we are looking for approximation schemes when deriving the relations within FMA, and we are investigating the usefulness of other relations in FMA like: X *is-attached-to* Y, X *sends-output-to* Y, etc. These relations will produce more matches different from those we present here.

Our findings were concluded from experiments conducted on medical test data. Therefore, we are conducting similar experiments in music domain on matching styles and genres from different music providers. In contrast to the medical domain, the knowledge in music is much weakly structured as different music content providers largely disagree on the meaning of music terms. In this direction, we took the effort to extract relatively extensive music ontology from Wikipedia which will serve as background knowledge in our experiments.

References

1. Jian, N., Hu, W., Cheng, G., Qu, Y.: Falcon-ao: Aligning ontologies with falcon. In: K-Cap 2005 Workshop on Integrating Ontologies. (2005)
2. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal **10**(4) (2001)
3. Ehrig, M., Sure, Y.: Foam - framework for ontology alignment and mapping; results of the ontology alignment initiative. In Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H., eds.: Proceedings of the Workshop on Integrating Ontologies. Volume 156., CEUR-WS.org (2005) 72–76
4. Aleksovski, Z., Klein, M., ten Kate, W., van Harmelen, F.: Matching unstructured vocabularies using a background ontology. In: Proceedings of Knowledge Engineering and Knowledge Management (EKAW). (2006)
5. Aleksovski, Z., Klein, M., ten Kate, W., van Harmelen, F.: Improving ontology mapping by using web ontologies as background knowledge. In: Submitted for acceptance at International Semantic Web Conference (ISWC). (2006)
6. de Bruijn, J., Martin-Recuerda, F., Manov, D., Ehrig, M.: D4.2.1 state-of-the-art-survey on ontology merging and aligning v1. SEKT Project deliverable D4.2.1 (2004)
7. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics **IV** (2005) 146–171
8. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-match: an algorithm and an implementation of semantic matching. In: Proceedings of the European Semantic Web Symposium (ESWC). (2004) 61–75
9. Hollink, L.: Enriching search queries for image retreival using wordnet. In: Submitted for acceptance at ISWC'06. (2006)

Towards understanding the needs of cognitive support for ontology mapping

Sean M. Falconer[1], Natalya F. Noy[2], and Margaret-Anne Storey[1]

[1] University of Victoria, Victoria BC V8W 2Y2, Canada
`{seanf, mstorey}@uvic.ca`
[2] Stanford University, Stanford, CA 94305, USA
`noy@stanford.edu`

Abstract. Researchers have developed a large number of ontology-mapping algorithms in recent years. However, ontology mapping is hardly a fully automated task and users must verify and fine-tune the mappings resulting from automated algorithms. Both academic and industry researchers have focused on the algorithms themselves and largely ignored the issue of cognitive support for users in the task of analyzing mappings proposed by the algorithms and creating new mappings. The lack of comprehensive user-oriented tools for ontology mapping (rather than just algorithms) hinders the adoption of the new technologie. In this paper, we analyze requirements for cognitive support for the ontology-mapping task. Recognizing that many researchers must focus on improving the algorithm performance itself (or only on providing better visualization), we have developed a plugin framework that enables developers to assemble a comprehensive ontology-mapping tool by plugging in various components. We provide a reference implementation of the complete framework. Thus, developers can plug in only the components they are interested in. For example, algorithm developers can plug in their algorithm and use the visualization components that we provide and the user-interface researchers can use the framework to experiment with various visualization paradigms for ontology mapping (and not worry about implementing the algorithms themselves). We also discuss specific cognitive aids for ontology mapping that we have developed and that are available as part of this framework.

1 Introduction

As ontologies become more commonplace and their number grows, so does their diversity and heterogeneity. Reconciling different ontologies and finding correspondences between their concepts is likely to be a problem for the foreseeable future. Thus, research on ontology mapping has become a prominent topic in the Semantic Web and ontology communities. There are mapping contests that compare the effectiveness of different algorithms [10], and researchers have proposed a standard mapping language [11]. As the results of the contests show, ontology mapping is far from being a fully automated task. We believe that in most cases manual intervention will be required to verify or fine-tune the mappings produced by the algorithms.

In general, a user interacting with an ontology-mapping tool, must examine the *candidate mappings* produced by the tool, indicate which ones are correct and which ones are not, and create additional mappings that the tool has missed. This process is a difficult cognitive task. It requires understanding of both ontologies being mapped and how they relate to each other. Also, both the ontologies themselves and the number of candidate mappings that the tools produce can be very large. However, there has been very little research on how to provide cognitive support for ontology mappings. Researchers have focused on improving the performance of the algorithms themselves, largely ignoring the issue of *end-user tools* (with a few exceptions [12, 18, 24]).

Perhaps, one of the reasons this issue has not been addressed is the lack of understanding of the importance of cognitive support [23]. Cognitive support measures how well a tool supports a user's cognitive processes, and it results from the interplay between the system image and the user's needs [8]. As Walenstein states, "The first rule of tool design is to make it useful; making it usable is necessarily second, even though it is a close second . . . [A tool's] usefulness is ultimately dependent upon [its] utility relating to cognition: i.e. to thinking, reasoning, and creating. Assistance to such cognitive work can be called cognitive support." [23]

Cognitive-support research is still new to software and knowledge engineering. In software engineering, generally, most tools are built with some consideration of utility and usability. Only recently, researchers started using a more formal approach to address these issues. The situation is similar in the field of knowledge-engineering tools. In this paper, we bring cognitive-support considerations to the ontology-mapping tools, and outline a set of requirements for these tools.

We believe that in order for the ontology-mapping tools to reach beyond research labs, both the performance of automatic ontology-mapping algorithms and the quality of cognitive support in ontology-mapping tools must improve. Recognizing that in many cases, researchers must focus on one or the other of these tasks, we have developed a plugin framework that covers many of the sub-tasks of ontology mapping, from specifying algorithms for initial comparison to executing the mappings. This framework is part of the PROMPT ontology-management suite [18], itself a Protégé plugin.[3] We have developed a reference implementation for each of the steps, including a number of cognitive aids. Developers can plug in their own components and use plugins developed by others (including our team) in order to fill in the missing pieces to have a comprehensive end-user tool.

This paper makes the following contributions:

- We analyzed requirements for cognitive support in ontology mapping (Section 2).
- We developed the PROMPT plugin architecture for ontology-management that enables developers to assemble a comprehensive ontology-mapping tool with their own components as part of the tool (Section 4).
- We implemented a set visualization plugins to the PROMPT plugin architecture that provides cognitive support for users in the ontology-mapping task (Section 5.2)

2 Requirements for Cognitive Support in Ontology Alignment

The set of end-user tasks in an ontology-mapping tool that we identified and the corresponding requirements are based on the common problems we have experienced and

---

[3] `http://protege.stanford.edu`

witnessed. We assume that the user's tasks involve verifying and fine-tuning the mappings produced by the automatic component and creating the mappings that the automatic algorithm missed. The following is a preliminary list of tasks that must be supported during the mapping process. Some of the requirements below address visual aids that make the user's job easier, and others are tasks that help reduce the user's cognitive load.

Navigation of ontologies being mapped: provide full access to the source and target ontologies.

Incremental navigation: enable browsing of the ontologies being mapped with the terms in the current mapping as focal points. Incremental navigation restricts the focus to the terms and allows the user to visually verify that the two terms suggested in the mapping have similar context and similar neighbors.

Identification of "candidate-heavy" ontology regions: identify visually which sections of the ontologies have large numbers of candidate mappings. Users may often want to focus on the sections where many of the mappings are, since these are likely to be the sections of the two ontologies where most of the mapping takes place.

Browsable list of candidate mappings: provide easy navigation and filtering of the candidate mappings produced by the automatic step. There must be a way for the user to instruct the tool to focus on certain regions of the ontologies or categorize the candidate mappings they are verifying. Such support allows the user to focus on smaller tasks and reduce complexity by validating higher priority matches first.

Information about the reasons a mapping was suggested: provide the user with some indication of why the automatic algorithm suggested a particular mapping. This reason helps establish trust between the user and the algorithm. For example, state that the two ontology terms matched exactly, or were synonyms of each other.

Context for mapping terms: display where the terms being mapped are in the ontology. Easy access to this information is essential in enabling the user to verify candidate mappings. In particular, the neighborhood of a term (immediate parent and children in the `is_a` hierarchy) may be especially important.

Definitions for mapping terms: provide easy access to full definitions of the terms in the ontology. For example, the definition might include the properties of a class and restrictions on those properties. Like the neighborhood, the internal structure helps explain the meaning of the term.

Conflict resolution and inconsistency detection: indicate to the user if some of the mappings that he has created produce conflicts or are inconsistent. Conflicts can arise from a variety of situations, such as when two concepts are mapped, but some structural elements that are critical for their definition have not been mapped yet.

Ability to save the verification state: The verification process must support potential interruptions where the user must be able to save their current progress and restart from that point at a later time.

Verification of mappings through execution: enable users to "execute" the mappings, for example, by transforming instances from the source to the target ontology based on specified mapping or using queries that access those individuals but this time direct them to the newly mapped term One can view such a transformation as a *debugging* step in creating a complete mapping: the user can verify if the instances created in the target from the source instances are the ones that he expected.

Direct creation and manipulation of the mappings: enable users to add details to the verified mappings. For example, a user may specify that a value of one property must be changed in a specific way in order for the mapping to be correct. Also, users may want to add metadata to mappings and describe their reasons for creating the mapping. The users will also often add mappings that the tool has missed.

Navigation of verified and manually specified mappings: One result of the user interaction with a mapping tool is a set of verified mappings, additional mappings, and details about the mappings. Users must be able to navigate this information.

Progress feedback: inform the user about their current progress in the mapping. How much they have verified and how much is left to verify. Verifying mappings can be a lengthy process, but providing feedback about the users' progress enables them to see that they are moving in the correct direction.

3 Related Work

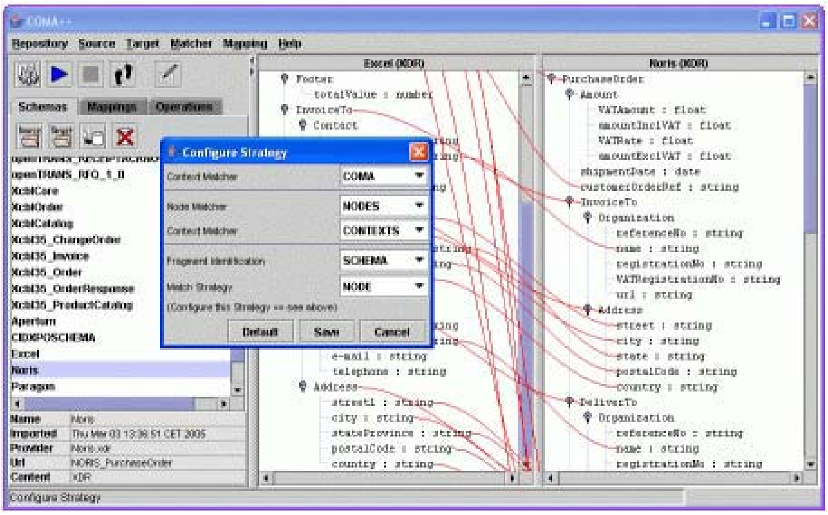
Cognitive support is about introducing artifacts in order to improve cognitive systems [23]. As Norman states [17], "The power of the unaided mind is highly overrated. Without external aids, memory, thought, and reasoning are all constrained." Although cognitive support can be addressed in a variety of ways, one popular approach is through *information visualization*. Information visualization leverages innate human abilities to perform spatial reasoning and make sense of relatively complex data using some form of graphical representation language [9]. Information visualization is often used to construct an advanced user interface to aid humans understand and navigate complex information spaces. In software engineering, this approach has been applied specifically to applications such as source code evolution [22] and algorithm animation [2].

Knowledge-engineering tools often use visualization to help users navigate ontologies. Usually, the problem these visualization tools are addressing is the comprehension and navigation of large information spaces. Different types of graph layouts are often used in order to display the ontology from different perspectives (e.g., see [21]). One of the goals of providing these various layouts is to help users view the same knowledge in different formats and potentially to validate and invalidate their mental models. As Richer and Clancy state, "…providing multiple views of the same knowledge or behavior can help a user understand a complex system." [19]

Navigation is also a relevant issue in ontology mapping, as users need to understand the structural context related to the match operations they must verify. However, in ontology mapping the focal point of the navigation is the terms involved in the mapping, not necessarily the entire information space.

3.1 Ontology-Mapping Tools and Their User Interfaces

Most user interfaces for mapping tools fall into one of three categories: graphical user interface, console-based, and finally web-based. Both the console-based and web-based tools follow similar approaches; the user supplies URIs for two ontologies, submits the input, and the tool processes the ontologies producing a list of potential matches. FOAM [7], MoA Shell [13], Chimaera [16], and the OWL Ontology Aligner [25] all



Context Matcher: COMA
Node Matcher: NODES
Context Matcher: CONTEXTS
Fragment Identification: SCHEMA
Match Strategy: NODE
(Configure this Strategy => see above)

Name: Noris
Imported: Thu Mar 03 13:36:51 CET 2005
Provider: Noris.xdr
Url: NORIS_PurchaseOrder
Content: XDR

Fig. 1: Mapping two schemas in COMA++.

fall into one of these two categories. Some of these tools, such as FOAM, also support interactive modes where a user can verify matches as they are computed.

Clio [12] and COMA++ [5] are examples of tools that support graphical user interfaces. The number of visual paradigms that the tools use to display the mappings is quite limited, however. Clio was developed by IBM for generating mappings between relational and XML schemas. Clio can infer correspondences in the source and target schemas and it also allows users to draw correspondences between parts of the schemas. Once the correspondences have been generated and verified, Clio generates queries to drive the translation from the source schema to the target schema. COMA++ works similarly, although it also supports ontology mapping. COMA++ automatically generates mappings between the source and target schemas, and draws lines between matching terms. Users can also define their own term matches (see Fig. 1). Both tools draw mappings between the source and target schemas, which can be difficult to work with when there are a lot of mappings or the distances between mapped terms is large.

These tools provide a mechanism to allow the user to supply an initial set of matches. This mechanism may be adapted to store a partially verified mapping and to restart verification at that particular stage. Also, tools like Clio and COMA++ support in-tool navigation of ontologies. There's also recently been an effort by a small number of researchers towards investigating applying visualization techniques to ontology alignment. AlViz [15], a plugin for Protégé, applies multiple-views via a cluster graph visualization along with synchronized navigation within standard tree controls. Generally, there is a dearth of visual paradigms for ontology mapping. Until we have such end-user tools with good cognitive support, many of the mapping algorithms that researchers develop, are unlikely to leave the labs. Hence, we have developed a plugin framework that treats an ontology-mapping process as a standard sequence of steps (from initial comparison of ontologies to executing mappings) and enables developers to substitute any of the steps with their own tools.

3.2 Component Frameworks for Ontology Mapping

Several researchers have addressed the issue of deomposing an ontology-mapping process into a sequence of subtasks–a step necessary for introducing a plugin framework for ontology mapping [6, 14]. These subtasks include pre-processing of the source ontology, configuration of the mapping algorithm, analysis of the results, and iterative invocation of the mapping algorithm. In a sense, implementing a plugin framework for ontology mapping starts with identification of this set of tasks. However, in order to enable developers to substitute implementation of any of the tasks with their own, we also must define interfaces, and must provide extension points in the tools.

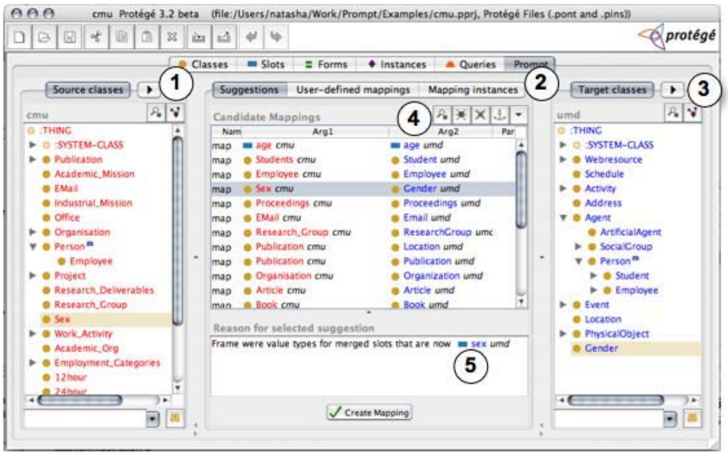
The work that is closest to our plugin framework for mapping is the IBM XML Mapping technology [20]. In this work, the authors developed a plugin framework for mapping data sources such as relational database schemas, UML models, and XML files to XML Schema. Their architecture distinguishes four core components, each of which has extension points for plugins: (1) user interface, with plugins for viewing different types of models; (2) mapping population, allowing developers to plugin different mapping algorithms; (3) mapping representation, enabling different forms of representing the mappings; and (4) code generation, providing runtime engines for executing the mappings. We take a similar approach in our work. However, we focus more on cognitive support for mapping, with user interface being a very prominent component. In addition, our work is applied to ontologies and not only to XML schemas.

4 PROMPT Plugin Architecture

PROMPT is a Protégé plugin that supports various tasks for managing multiple ontologies, including ontology mapping [18]. In its original form, PROMPT starts the ontology-mapping process by performing initial comparison of the source and target ontologies to be mapped, mainly based on lexical comparison of class names. After the initial comparison, PROMPT presents the user with a set of candidate mappings. A user can examine the mappings, create new mappings, and save the correct ones. As the user identifies a mapping as correct, PROMPT performs structural analysis of the neighborhood of the mapped concepts, suggesting new mappings based on the graph structure. PROMPT saves the mappings as instances in the *mapping ontology* [3]. After the user defines the mappings, he can run a *mapping interpreter* [4] to transform instances from the source to the target ontology based on the mapping.

When verifying candidate mappings, users can access the source and target ontologies in the Protégé interface; when a user selects a mapping to examine, the corresponding concepts are highlighted in the ontologies trees. A user can also navigate to the tab that displays the mapping ontology and its instances and edit the mappings there directly (see Fig. 2).

The PROMPT plugin framework allows developers to replace any of the components that we have just described with their own. The plugin framework works by providing Java interfaces for various types of plugins (comparison algorithm, visualization components, etc). A plugin developer chooses the interface they wish to implement, and then supplies the appropriate method bodies in order to perform the operations they wish to


cmu Protégé 3.2 beta (file:/Users/natasha/Work/Prompt/Examples/cmu.pprj, Protégé Files (.pont and .pins))

Candidate Mappings

Nam	Arg1	Arg2	Par
map	age cmu	age umd	
map	Students cmu	Student umd	
map	Employee cmu	Employee umd	
map	Sex cmu	Gender umd	
map	Proceedings cmu	Proceedings umd	
map	EMail cmu	Email umd	
map	Research_Group cmu	ResearchGroup umc	
map	Publication cmu	Location umd	
map	Publication cmu	Publication umd	
map	Organisation cmu	Organization umd	
map	Article cmu	Article umd	
man	Book cmu	Book umd	

Reason for selected suggestion

Frame were value types for merged slots that are now sex umd

Fig. 2: The PROMPT user interface and the extension points in PROMPT's mapping component. The left column shows the source ontology; the middle column displays the mappings suggested by PROMPT and explanations of these suggestions. The right column displays the target ontology. There are tab extensions points for the source (1), mapping (2), and target (3) components. Area (4) shows the suggestion header button extension point. Algorithms can provide their own explanations for each candidate mapping (5).

execute. More specifically, we view the ontology-mapping process as a sequence of the following steps (Fig. 3):

Perform initial comparison of the ontologies: an algorithm compares two ontologies and produces a list of candidate mappings.

Present candidate mappings to the user enabling him to analyze the results. This step includes components for cognitive support (various visualizations of the source and target ontologies, options to filter content presented in the display, etc.) and interactive comparison algorithms that are invoked either explicitly by the user or as a result of mappings being verified.

Fine tune and save the mappings in a declarative mapping format.

Execute mappings to transform instances from source to target or to perform other operations.

In the current implementation, developers can replace components of any of the steps in this list, and our plan is to make all of the steps replaceable.

Figure 4 shows the PROMPT screen for configuring an algorithm for initial comparison. The user has chosen to run a FOAM algorithm at this stage. The integration of FOAM and PROMPT is available as part of PROMPT distribution [4]. A developer of an algorithm plugin can specify not only how to invoke the algorithm, but also how the configuration screen presented to the user should look like.

User-interface extension points exist throughout the PROMPT mapping interface. We currently support extensions that allow a developer to add new tabs to the source,

---

[4] At this time, the only algorithms integrated are FOAM and the original PROMPT algorithms.

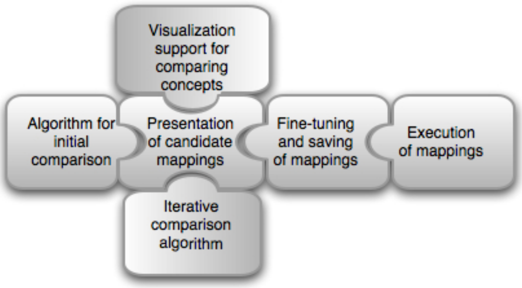
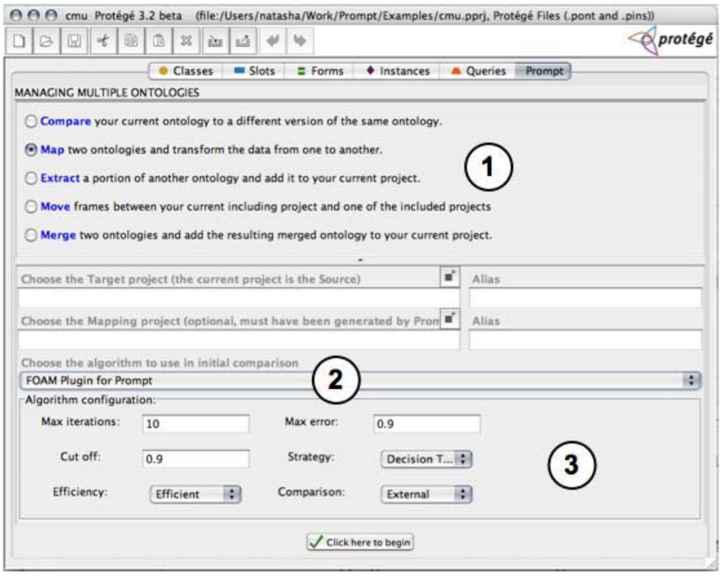


Fig. 3: Configurable steps in the PROMPT framework. Developers can replace any component in the figure with their own implementation. Cognitive aids can be applied at each step to ease cognitive load.



cmu Protégé 3.2 beta (file:/Users/natasha/Work/Prompt/Examples/cmu.pprj, Protégé Files (.pont and .pins))

MANAGING MULTIPLE ONTOLOGIES

○ **Compare** your current ontology to a different version of the same ontology.
● **Map** two ontologies and transform the data from one to another.
○ **Extract** a portion of another ontology and add it to your current project.
○ **Move** frames between your current including project and one of the included projects
○ **Merge** two ontologies and add the resulting merged ontology to your current project.

Choose the Target project (the current project is the Source) | Alias
Choose the Mapping project (optional, must have been generated by Prom | Alias

Choose the algorithm to use in initial comparison
FOAM Plugin for Prompt

Algorithm configuration:

Max iterations: 10
Max error: 0.9
Cut off: 0.9
Strategy: Decision T...
Efficiency: Efficient
Comparison: External

Fig. 4: Selecting an algorithm in PROMPT. The area marked (1) shows the options in PROMPT, here we have selected to map two ontologies. The area marked (2) displays the algorithm plugins available, here we have selected our custom built FOAM plugin. Finally, area (3) shows the algorithm configuration panel supplied by the FOAM plugin.

mapping, and target panels in the mapping component (see Fig. 2). Also, new header buttons can be added to the mapping suggestion list. For example, developers could use actions on these buttons to filter information (Section 5.2).

5 Cognitive Support in PROMPT

We examine issues of cognitive support of the core PROMPT plugin (before the addition of the interface plugins discussed in Section 5) as an example of questions that arise when developing support for such a cognitively complex task as ontology mapping. PROMPT already addresses many of the cogntive support requirements that we discussed in section 2. However, PROMPT may not address all of these requirements so that users can make use of the features effectively. Meeting the requirements is not enough, we must also understand the usability of the implemented features.

5.1 Potential Problems in a Mapping User Interface

Although we have not yet carried out formal user studies to understand PROMPT's usability, we have identified some questions that such usability testing will reveal. For example, when a user selects a candidate mapping, PROMPT highlights the terms involved in the mappings in the source and target display. Thus, the user can see the context of the terms in the mapping. But this feature could potentially introduce new cognitive issues. For example, the selection of the terms in the ontologies is immediate, the ontology trees are expanded directly to the term that needs to be displayed, no animation of this process exists. Does jumping immediately to the term destroy the user's global context about where they are in the two ontologies? Does it interrupt the user's work flow? What if the user had used the ontology tree to browse to a particular location, but selecting a suggestion removed the user's selected focal point?

Another related issue is incremental navigation in PROMPT. Currently, the user can browse the source and target ontologies via a tree control in the mapping component. However, selecting a suggestion immediately switches the tree's focus. Also, this type of view can be difficult to use when viewing items deep in the hierarchy.

PROMPT loads all its generated mapping operations into a browsable suggestion list. With large ontologies, this list could be very large. There is currently no support for sorting, categorizing, or filtering the list. Abrams and colleagues [1] found that web browser users will not put more than 35 items in their favorite's list before resorting to categorizing links within hierarchies or stopping their use of favorites all together. Similar issues may need to be addressed in PROMPT. For example, what will users do when presented with a list of a thousand or even a hundred suggestions?

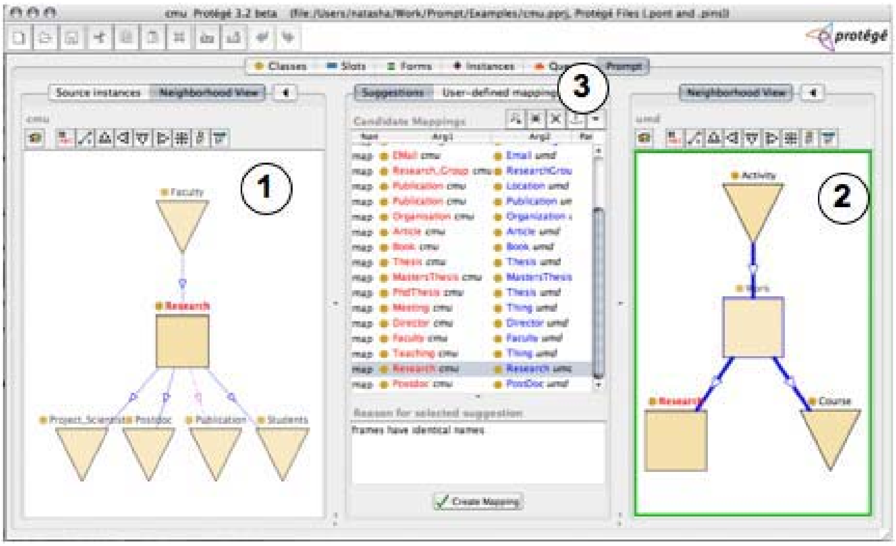
The final issue with PROMPT we wish to discuss is browsing the resulting mappings. Unlike Clio and COMA++, which draw lines between matching terms, PROMPT takes a different approach. Firstly, after the user confirms a mapping suggestion, the corresponding terms in both the source and target ontologies get a "mapped" icon associated with them indicating that the terms have already been mapped. Secondly, the occurance of the mapping event is recorded in a mapping ontology that is browsable by the user. Although the icon indicator certainly is less cluttered than the line drawing in Clio and COMA++, there is no explicit way for the user to visualize what the corresponding matched term is.

5.2 COGZ Interface Plugin for PROMPT

In order to address some of the missing requirements in PROMPT, we have developed COGZ—a user-interface plugin for the mapping component. COGZ attempts to provide user support for reducing the size and complexity of a mapping and to improve user interaction with establishing the term context and improving incremental navigation.

We addressed the complexity and size requirement by adding filters to the list of candidate mappings. There are several types of filters. First, users can filter candidate mappings based on the explanation provided by the algorithm that generated the candidate (see area (5) in Fig. 2). For example, the user can filter the list to inspect only exact term matches first, and then address more complex matches, like synonym or shared-hierarchy matches. There is also a filter that allows users to restrict mappings to classes from certain subtrees in the ontologies. Users can specify multiple subtrees in both the source and the target ontologies. This filter provides a powerful means for the user to address areas of the ontology they are most familiar with.

Finally, to address the context and navigation requirements, we added a *Neighborhood* tab to both the source and target ontology components. The neighborhood tab is synchronized with the browsing of candidate mappings: selecting a mapping displays the corresponding term's neighborhood. The neighborhood consists of the immediate parents and children of the term. The viewer supports incremental navigation by allowing the user to expand incrementally the neighborhood of any visible node. Also, the plugin provides six different layouts to allow the user to view the graph from a multitude of perspectives (see Fig. 5). The visualizations are provided by Jambalaya [21].



Type	Arg1	Arg2
map	EMail cmu	Email umd
map	Research_Group cmu	ResearchGrou
map	Publication cmu	Location umd
map	Publication cmu	Publication ur
map	Organization cmu	Organization
map	Article cmu	Article umd
map	Book cmu	Book umd
map	Thesis cmu	Thesis umd
map	MastersThesis cmu	MastersThesis
map	PhdThesis cmu	Thesis umd
map	Meeting cmu	Thing umd
map	Director cmu	Director umd
map	Faculty cmu	Faculty umd
map	Teaching cmu	Thing umd
map	Research cmu	Research umd
map	Postdoc cmu	PostDoc umd

Frames have identical names

Fig. 5: PROMPT's mapping component with the interface plugin. Areas (1) and (2) show the neighborhoods of the source and target terms of the currently selected suggestion. Area (3) shows the location of the filter button. The neighborhood display makes it clear that the two classes have different meaning in the two ontologies even though their names are the same ($Research$).

6 Discussion and Future Work

We have discussed the need for cognitive support in ontology mapping tools. We proposed several requirements for satisfying this need, which were based on our own background work and experience. While these requirements are preliminary, we believe they represent a good initial description of the problems faced by users performing mappings. We believe it is very important to refine these requirements by carrying out studies and surveys in the knowledge engineering community. Specifically, we plan to evaluate the effectiveness of PROMPT through user studies and tool usage statistics. We also plan to enhance the visualization plugin in order to further address the requirements that PROMPT does not fullfill.

We also discussed the implementation of a plugin framework for PROMPT. The framework helps address two fundamental issues. Firstly, how can we satisfy the cognitive support requirements in one consistent environment, and secondly, how can we close the gap between mapping algorithm research and mapping users. By supporting user interface extension points in PROMPT, experts and developers in Human Computer Interaction can incorporate their ideas and tools to help decrease the cognitive load on end users of mapping tools. Similarly, the algorithm extension points also help the algorithm researcher. By using these extensions, researchers (or software developers) can easily incorporate their algorithms into PROMPT, allowing the research to be available under one consistent user interface. End users will benefit from having access to the best known algorithms, as well as the best cognitive support tools available.

In addition to developing better cognitive support for mappings, other research challenges remain. Our decomposition of the mapping process (Fig. 3) may not be general enough. It does not account fully for comparison algorithms that require an initial set of mapped terms as input. One can invoke such an algorithm at the iterative step, but more direct support for specifying the inputs precisely will likely be required. Similarly, PROMPT assumes a declarative representation of mappings (mainly as instances in an ontology). We would like to extend it to allow the use of an alignment API (e.g. [11]) and in-memory access to mappings. We envision that as developers begin to use the plugin framework, we will need to introduce other extension points of this type.

We plan to further enhance the plugin framework by adding more extension points for algorithms and interface components. PROMPT, FOAM and COGZ plugins are available as part of the full installation of Protégé 3.2beta.[5] Instructions for plugin developers and additional information are available on the PROMPT wiki site.[6]

Acknowledgements

This work was supported in part by the National Center for Biomedical Ontology, under roadmap-initiative grant U54 HG004028 from the National Institutes of Health.

---

[5] `http://protege.stanford.edu/download/download.html`

[6] `http://protege.cim3.net/cgi-bin/wiki.pl?Prompt`

References

1. D. Abrams, R. Baecker, and M. H. Chignell. Information archiving with bookmarks: Personal web space construction and organization. In *Human Factors in Computing Systems (CHI 98)*, pages 41–48. ACM Press, 1998.
2. M. D. Byrne, R. Catrambone, and J. T. Stasko. Evaluating animations as student aids in learning computer algorithms. *Computers & Education*, 33(4):253–278, 1999.
3. M. Crubézy and M. A. Musen. Ontologies in support of problem solving. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 321–342. Sringer, 2003.
4. M. Crubézy, Z. Pincus, and M. Musen. Mediating knowledge between application components. In *Workshop on Semantic Integration at ISWC-2003*, Sanibel Island, FL, 2003.
5. H.-H. Do. *Schema Matching and Mapping-based Data Integration*. PhD thesis, Department of Computer Science, Universitt Leipzig, 2006.
6. M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In *1st European Semantic Web Symposium*, Heraklion, Greece, 2004. Springer, LNCS.
7. M. Ehrig and Y. Sure. FOAM–framework for ontology alignment and mapping; results of the ontology alignment initiative. In *Proceedings of the Workshop on Integrating Ontologies*, volume 156, pages 72–76, October 2005.
8. N. Ernst. Towards cognitive support in knowledge engineering: An adoption-centred customization framework for visual interfaces. Master's thesis, University of Victoria, 2004.
9. N. A. Ernst, M.-A. Storey, and P. Allen. Cognitive support for ontology modeling. *International Journal of Human-Computer Studies*, 62(5):553–577, 2005.
10. J. Euzenat. Eon ontology alignment contest. http://oaei.inrialpes.fr/2004/Contest/.
11. J. Euzenat. An api for ontology alignment. Technical report, 2006.
12. M. A. Hernandez, R. J. Miller, L. M. Haas, L. Yan, C. T. H. Ho, and X. Tian. Clio: A semi-automatic tool for schema mapping. In *SIGMOD Record*, 2001.
13. E. . T. R. Institute. Moa shell. http://mknows.etri.re.kr/moa/docs/moashell.html, 2003.
14. Y. Kalfoglou and M. Schorlemmer. IF-Map: an ontology mapping method based on information flow theory. *Journal on Data Semantics*, 1(1):98–127, Oct. 2003.
15. M. Lanzenberger and J. Sampson. Alviz - a tool for visual ontology alignment. In *IV '06: Proceedings of the conference on Information Visualization*, pages 430–440, Washington, DC, USA, 2006. IEEE Computer Society.
16. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. The chimaera ontology environment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 1123–1124. AAAI Press / The MIT Press, 2000.
17. D. A. Norman. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley, 1993.
18. N. F. Noy and M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
19. M. Richer and W. J. Clancy. Guidon-watch: A graphic interface for viewing a knowledge-based system. *IEEE Computer Graphics and Applications*, 5(11):51–64, 1985.
20. M. Roth, M. Hernandez, P. Coulthard, L. Yan, L. Popa, and C. Salter. Xml mapping technology: Making connections in an xml-centric world. *IBM Systems Journal*, 45(2):389–409, 2006.
21. M.-A. Storey, N. F. Noy, M. Musen, C. Best, R. Fergerson, and N. Ernst. Jambalaya: an interactive environment for exploring ontologies. In *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 239–239, New York, NY, USA, 2002.
22. L. Voinea, A. Telea, and J. J. van Wijk. Cvsscan: visualization of code evolution. In *SoftVis '05: Proceedings of the 2005 ACM symposium on Software visualization*, pages 47–56, New York, NY, USA, 2005. ACM Press.
23. A. Walenstein. *Cognitive support in software engineering tools: A distributed cognition framework*. PhD thesis, Simon Fraser University, Vancouver, BC, 2002.
24. D. Wenke, S. Atanassov, D. Manov, M. Maier, and W. Sperling. D4.5.1 report on ontology mediation as service component. Technical report, EU-IST SEKT Deliverable, 2003.
25. A. V. Zhdanova. Owl ontology aligner. http://align.deri.org:8080/deri/align.jsp.

Applying an Analytic Method for Matching Approach Selection

Malgorzata Mochol[1], Anja Jentzsch[1] and Jérôme Euzenat[2]

[1] Freie Universität Berlin, Institut für Informatik
Takustr. 9, D-14195 Berlin, Germany
`mochol@inf.fu-berlin.de`, `anja@anjeve.de`
[2] INRIA Rhône-Alpes
655 avenue de l'Europe, 38330 Montbonnot Saint-Martin, France
`Jerome.Euzenat@inrialpes.fr`

Abstract. One of the main open issues in the ontology matching field is the selection of a current relevant and suitable matcher. The suitability of the given approaches is determined w.r.t the requirements of the application and with careful consideration of a number of factors. This work proposes a multilevel characteristic for matching approaches, which provides a basis for the comparison of different matchers and is used in the decision making process for selection the most appropriate algorithm.

1 Introduction

Many methods and tools are under development to solve specific problems in the Semantic Web; however none of these solutions can be deployed for all problems in this area. This statement is also true in the ontology matching field, in which there is no and will never be an overarching matching algorithm for ontologies that is capable of serving all (heterogeneous) ontological sources. Most of the research in this area proposes new approaches based on different principles and relies on various features. These new approaches only solve small parts of "global" problems in the matching field or fill some open matching gaps[12]. Therefore in general, when implementing an application using a matching approach, the corresponding algorithm is typically built from scratch and no attempt to reuse existing methods is made. Despite an impressive number of research initiatives in the matching field, containing valuable ideas and techniques, current matching approaches still feature major limitations when applied to the emerging Semantic Web. For example, the majority of existing approaches to ontology matching are (implicitly) restricted to processing particular classes of ontologies and thus they are unable to guarantee a predictable quality of results on arbitrary inputs. What is required are appropriate ontology matching techniques capable of coping with different levels of detail in concept descriptions[3]. Aside from the problems mentioned above, there are many other open issues, of a global nature, which need to be solved in the future. Firstly there is the question of what should be matched based upon what needs to be found. Also it is important to avoid performing relatively blind matching, while being aware of when to stop the matching process. Furthermore, the selection of a currently relevant matching algorithm that is suitable w.r.t the given specification and the definition of the appropriate criteria for this decision making process needs to be taken into account. Regarding the latter, one of the first steps on the way to solve this issue can be an infrastructure for taking advantage of existing ontology alignments. To tackle the issues of the heterogeneity of existing ontology matchers as well as to limit the disadvantages of the singular approach a *reuse strategy* for matching approaches based on the examination of their characteristics is needed. The first goal within such a methodology is the detection of potentially suitable approaches from the huge number of existing methods. After the analysis of existing approaches, evaluation of their usage context and conducting interviews with various domain (matcher) experts some factors were identified that are relevant for the selection of a suitable matcher(s) w.r.t the requirements of the given application. The objective of the work is to develop a framework that takes into account the characteristics of matching algorithms and offers methods and tools to support the process of selecting an applicable matcher. The rest of this paper is organized as follows: Section 2 gives an overview of the relevant criteria to describe and compare matching approaches. This is followed by a description of one of the methods for multi-criteria decision making called Analytic Hierarchy Process (AHP) in Section 3 and its application into a matching selection process in Section 4. The conclusions, along with future work, are discussed in Section 5.

2 How to Characterize Matching Approaches?

One of the main open issues in the ontology matching field is that of choosing a *current relevant* and *suitable* matching algorithm. Since there is no such thing as “general” matching problem, there is thus no “general” way to solve the matching issues by only posing the query “find a matching algorithms for two ontologies and deliver a set of relations”. This query covers indeed ever type of ontology and matching algorithm it also gives same basics information about the alignments however it does not address the specific requirements of a particular application. The matching algorithm should not only be chosen with respect to the given data but should also be adapted to the system, taking into consideration the problem to be solved by the approach, for example merge ontologies to create new one, match ontologies to compare profiles, match data etc.

The matching problem should be seen as a collection of small particular subproblems, which are dependent on various criteria and circumstances. Following this idea, for a given (characterized) pair of ontologies to be matched, having a definition of the problem to be solved along with particular requirements regarding the final application, one must decide which matching algorithms are to be applied to satisfy these specification and to obtain the desired output. Possible attributes, that could have an impact on the selection of an adequate matching approach, must be defined in order to find a suitable solution to this issue. Accounting for the empirical findings of different case studies in ontology engineering[23–25], and regarding the requirements collected during the development of different Semantic Web application scenarios[2, 13][3], as well as during the intensive collaborations with ontology and software engineers, six groups of factors (dimensions) has been defined as relevant for the matching selection process. These dimensions are the main aspects that must be taken into account during the examination of the suitability of a single matching approach for the solving of a given problem: (i)*input characteristic* that takes into account the ontologies to be matched; (ii)*approach characteristic* describes the matching algorithms themselves; (iii)*output characteristic* defines the desired result of the matching execution; (iv)*usage characteristic* takes into account the different situations where the approaches have been used; (v)*documentation characteristic* points out the existence and type of the documentation; and (vi)*cost characteristics* addresses the costs which have to be paid for the usage of the algorithm. The dimensions form the superficial collection for matcher attributes and build the first level of the so called *multilevel characteristic for matching approaches*. The multilevel characteristic is organized in the form of a taxonomy where dimensions are defined by sets of factors and these are described by the attributes. These characteristics can be illustrated as a hierarchical tree (cf. Fig 1) where the child nodes describe and represent the parent nodes' properties[20].

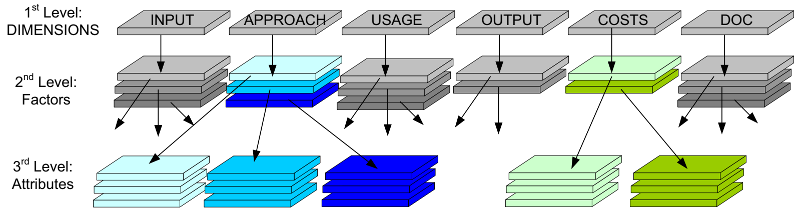


Fig. 1. Multilevel characteristic with dimensions, factors and attributes

In the following sections we briefly describe some of the factors of each dimension and state others in the form of tables (cf. Tab. 1,2,3,4,5,6) since the exact specification of all criteria would go beyond the scope of this paper.

2.1 Input Characteristic

The first step towards the analysis of the matching characteristics is the examination of the matching input. In our opinion, the attributes that describe the input are the most important and relevant criteria that play a crucial role in the selection of the appropriate algorithm. Despite the relatively large number of promising matching approaches their limitations w.r.t. certain ontology characteristics have often been emphasized in recent literature[14, 21, 22, 29, 30]. The dimension *input characteristic* describes not only the heterogeneity of the sources that are to be matched, e.g. *size* (some matchers perform well on relatively small inputs), *natural language* used for the definition of concepts (some algorithms require certain nat. language) and *input structure* (some matchers do

---

[3] Projects:(i)Wissensnetze,`http://wissensnetze.ag.nbi.de`,(ii)Reisewissen, `http://reisewissen.ag.nbi.de`, (iii)SWPatho, `http://swpatho.ag.nbi.de`,(iv)Knowledge Web `http://knowledgeweb.semanticweb.org/`

not perform well on heterog. structures[14]), but also takes into account *external sources*, which a matching algorithm can use for its execution (cf. Tab. 1).

DIMENSION: INPUT CHARACTERISTIC	
Factor	**Description**
Input Size (algorithm is able to handle:)	
number of ontologies	number of different ontologies to be matched (two or more)
size of input	number of ontological primitives (concepts,properties, axioms, instances) to be matched: *small* (up to 100 primitives), *middle* (100-1000 primitives) *big* (over 1000 primitives)
size of instances	number of instances to be matched: *no instances*, *small* (up to 100 primitives), *middle* (100-1000 primitives), *big* (over 1000 primitives)
number of concepts	number of concepts to be matched: *small* (up to 100 primitives), *middle* (100-1000 primitives), *big* (over 1000 primitives)
number of relations	number of relations to be matched: *small* (up to 100 primitives), *middle* (100-1000 primitives), *big* (over 1000 primitives)
number of axioms	number of axioms to be matched: *no instances*, *small* (up to 100 primitives), *middle* (100-1000 primitives), *big* (over 1000 primitives)
Input category (algorithm is able to handle:)	
glossary	a list of terms with the definitions for those terms
thesaurus	a list of important terms (single-word or multi-word) in a given domain and a set of related terms for each term in the list
taxonomy	indicates only class/subclass relationship (hierarchy)[9]
DBschema	often does not provide explicit semantics for their data
ontology	an explicit specification of a conceptual.[16]; describes a domain completely[9]
Input formality level[32, 33] (algorithm is able to handle:)	
(highly/semi) informal ontology	expressed loosely in natural language or in a restricted and structured form of natural language
semi-formal ontology	expressed in an artificial formally defined language
(rigorously) formal ontology	meticulously defined terms with formal semantics, theorems and proofs of such properties as soundness and completeness
Input model type (algorithm is able to handle:)	
task ontology	model build for a specific task
application ontology	model build for a specific application
domain ontology	model of a specific domain or part of the world
upper-level ontology	model of the common objects that are generally applicable across a wide range of domain ontologies; it describes very general concepts
Input type (algorithm is able to handle:)	
scheme	schema-based matcher
instance	instance/contents-based matchers
External sources (algorithm is able to handle /to provide:)	
additional user input	most matchers rely not only on the input to be matched (like schemas or instances) but also on auxiliary information
previous matching decision	
training matches	
domain constrains	
list of valid domain values	
dictionary	
miss-match information	
matching rules	
global schemas	
Input natural language (NL) (algorithm is:)	
NL-specific (one language)	the approach is dependent on one natural language
NL-specific (many languages)	the approach is dependent on more then one natural languages
NL-independent	the approach is language independent
Input representation language (RL)[33] (algorithm is:)	
RL-specific (one language)	the approach is dependent on one rep. language
RL-specific (many languages)	the approach is dependent on more then one rep. languages
RL-independent	the approach is independent on rep. language
Input structure (algorithm is able to handle:)	
tree structure	the approach can handle only tree-structers
graph structure	the approach can handle (heterogenous) graph structers
is-a relations	the approach can handle is-a relations
heterogeneous relations	the approach can perform not also on heterogeneous relations

Table 1. Input characteristic

2.2 Approach Characteristic

The second crucial dimension characterizes the matching approaches themselves. The corresponding factors and attributes compile a list of matcher features that are empirically proved to have an impact on the quality of matching tasks. They consider e.g. the common classification of the approaches[5, 26, 29] and distinguish between *individual algorithms*[14, 31] and combinations of the individual algorithms: *hybrid* and *composite solutions*. A hybrid approach[21] follows a *black box paradigm*, in which various individual matchers are synthesized into a new algorithm, while the composite matchers allow an increased user interaction[6, 8].

The *approach characteristic* also takes into account issues like processing type, matching ground and execution parameter (cf. Tab. 2).

DIMENSION: APPROACH CHARACTERISTIC	
Factor	**Description**
Matcher Type (algorithm is a(n):	
individual matcher	computes a mapping based on a single matching criteria
combined matcher	uses multiple individual matchers
Processing (algorithm supports:)	
manual execution	manual execution
white box paradigm	semi-automatic execution where the human intervention is possible
black box paradigm	automatic execution without human intervention
manual preprocessing allowed / required	human intervention before the execution is allowed or even required
manual postprocessing allowed /required	human intervention after the execution is allowed or even required
simultaneous execution	the single matching algorithms (within a composite matcher) can be executed simultaneously
sequential execution	the single matching algorithms (within a composite matcher) can be executed sequentially
Kind of Similarity Relation (algorithm performs:)	
syntactic matching	similarity based on syntax driven techniques and syntactic similarity measures; relation computed between labels at nodes[29]
semantic matching	relation computed between concepts at nodes[29]
Matcher Level (algorithm can perform on:)	
element level	match performed for individual schema elements
structure level	match performed for complex schema structures
atomic level	elements at the finest level of granularity are considered e.g. attributes in an XML schema[26]
non-atomic (higher) level	e.g. XML elements
Matching Ground	
heuristic	"guessing" relations between similar labels or graph structures[28]
formal	uses formal techniques (e.g. can have model-theoretic semantics which is used to justify the results)[28]
Semantic Codification Type(algorithm uses:)	
implicit techniques	syntax driven techniques[28](e.g. considers labels as strings)
explicit techniques	exploit the semantics of labels[28]; uses an external sources for assessing the meaning of labels
Execution Parameter (algorithm needs:)	
max time of execution	describes the maximal needed time of execution
max disc space for execution	describes the maximal needed disc space
precision	expresses the proportion of retrieved matches which are relevant[34]
recall	expresses the proportion of relevant documents retrieved[34]

Table 2. Approach characteristic

2.3 Usage Characteristic

One of the fundamental requirements for the realization of the vision of the fully developed Semantic Web are "tried and tested" ontology matching algorithms. Though containing valuable ideas and techniques some of the current matching approaches lack exhaustive testing in real world scenarios. Considering this problem and additionally making allowance for the fact that some of the algorithms cannot be applied across various domains to the same effect[14], it is important to know, if a particular approach has already been successfully adapted for different *domains*, *applications* and *tasks*. Additionally, the *usage characteristic* dimension also considers *different types of users*: ontology engineers who e.g. look for means to compare sources for building a new ontology or Web Services seeking automatized methods to generate mediation ontologies (cf. Tab. 3).

2.4 Output Characteristic

In addition to the input, approach and usage dimensions, the *output characteristic* (cf. Tab. 4) plays a decisive role in the process of selecting the suitable matching algorithm. Depending on the given requirements, an application can for example need a matcher that considers only some of elements of the schemes, while other systems might lack a match for all elements. One of the key factors in this dimension is the *cardinality* (global vs. local cardinality) which specifies whether a matcher compares one or more elements of one scheme with one or more elements of another scheme (in some cases the results are based on a one-to-one mapping between taxonomies[7] and in others on one-to-n).

DIMENSION: USAGE CHARACTERISTIC	
Factor	**Description**
Usage goal (algorithm is build for:)	
local use	approach developed for local use
network use	approach developed for network use
internet-based use	approach developed for internet-based use
Application Area (algorithm is build for:)	
reuse of sources	the matching approach is applied to ontology reuse which may be defined as a process in which available knowledge is used as input to generate new ontologies
usage of sources	the matching approach is applied to use the ontologies (within an application) e.g. to compare profiles
integration	reusing available source ontologies within a range to build a new ontology which serves at a higher level in the application than that of various ontologies in ontology libraries[19]
translation	ontology translation is required when translating data sets, generating ontology extensions, and querying through different ontologies[10]
Usage type (algorithm is:)	
applicable by human	approach can be used only by humans (human interaction indispensable)
applicable by machine	approach can be used by machine as a service
Adaption parameter (algorithm has been applied for:)	
number of domains	number of different domains the matching approach was applied for
number of applications	number of different applications the matching approach was applied for
number of tasks	number of different tasks the matching approach was applied for
reference of usage	has the approach been utilized by other users

Table 3. Usage characteristic

DIMENSION: OUTPUT CHARACTERISTIC	
Factor	**Description**
Output type	
deliver relations	the output of most matching systems is a set of the correspondences between attributes of schemas
deliver value	e.g. matcher used to determine the semantic similarity between concepts
deliver understandable	matcher delivers some explanations of the results
(for humans) results	
Matching Cardinality	
global 1:1	relationship cardinalities between matching elements w.r.t different mapping elements[26]
global n:1	
global 1:m	
global n:m	
local 1:1	relationship cardinalities between matching elements w.r.t an individual mapping element[26]
local n:1	
local 1:m	
local n:m	
Execution Completeness	
full match	considers all elements of the schemes
partial match	considers only some elements of the schemes
injective match	all elements of the domain are mapped to elements of the range
surjective match	all elements of the range are mapped to elements of the domain

Table 4. Output characteristic

2.5 Documentation Characteristic

Due to the fact that documentation is an essential part of every software product and in many ways it is even more important than the program code[18] the information about its quality and clarity can be significant for the selection of an approach. Furthermore, since one of the goals of documentation is to provide sufficient information so that an architecture can be analyzed for suitability to the purpose[4], it could be a determining coefficient for the selection of a particular algorithm, especially if the algorithm is to be reused in a different context from the domain or application it was originally developed for (cf. Tab. 5).

2.6 Cost Characteristic

The last dimension, *cost characteristic*, describes the financial factors regarding the (commercial) usage of a single matching approach like the matcher licence or the access to the appropriate matcher interface (cf. Tab. 6).

DIMENSION: DOCUMENTATION CHARACTERISTIC	
Factor	**Description**
quality of documentation	quality of the available documentation
clarity of documentation	clarity of the available documentation
clarity of maturity description	clarity of the description of the approach's maturity
availability of examples	are examples of the approach available

Table 5. Documentation characteristic

DIMENSION: COST CHARACTERISTIC	
Factor	**Description**
costs of matcher licence	the costs that have to be paid for the matcher licence
costs of matcher tool licence	the costs that have to be paid for the using of the tools matcher have been developed with
costs of access matcher interface	the costs that have to be paid for the using of interface

Table 6. Cost characteristic

3 A Method to Detect Suitable Matching Approaches

In the previous section we introduced the multilevel characteristic for matching approaches that provides a framework for matcher description. It can be used as a basic principle in the process of comparing different algorithms to determine an appropriate approach w.r.t the given circumstances.

As long as decisions rely on single criterion that serves as the basis for comparison of alternatives or the scales of the different criteria are consistent and numeric measures accurately capture expected performance, summary statistics or, in some cases, just acting on the human instinct may be sufficient for the decision making process. However, when the decision depends on multiple criteria and scales are not consistent the process becomes very complex and difficult, and the involvement of qualitative as well as quantitative methodologies or tools is indispensable. Consequently, in such cases a multi criteria decision making process is required, otherwise known as a *Multi Criteria Decision Analysis (MCDA)*, which is a procedure that aims to support decision makers whose problems are concerned with numerous and conflicting criteria. Such methods developed for better model decision scenarios vary in their mathematical rigor, validity, and design[15]. One of such method, a methodology for supporting a decision making process called *Analytic Hierarchy Process (AHP)* takes into account the considerations of Hahn[17] regarding the need for a structured results-based approach for decision making that allows trade-offs into the systematic method, including all perspectives and considerations. The AHP is a systematic approach developed to structure the expectance, intuition, and heuristics based decision making into a well-defined methodology on the basis of sound mathematical principles[1]. It helps to set priorities and to make the best decision when both qualitative and quantitative aspects of a decision need to be considered[27], i.e. AHP provides a mathematically rigorous application and proven process for prioritization and decision-making. By reducing complex decisions to a series of pair-wise comparisons and then synthesizing the results, decision-makers arrive at the best decision based on a clear rationale. It is generally accepted, that AHP constitutes one of the best options to aid multi-criteria decision making since it does not use the normalized groups of separate numbers which destroy the lineal relationship among them[11]. Instead it compares the relative importance that each

criterion has with respect to the others, while enabling the relative weight of the criteria to be calculated. Finally it normalizes the weights in order to obtain the measures for the existing alternatives. The AHP-method consists of:
STEP 1 - define the problem or the project objectives: e.g. buying a car;
STEP 2 - build a hierarchy of decision: AHP provides a means to break down the problem into a hierarchy of subproblems (hierarchy of goal, criteria, sub-criteria and alternatives) which can more easily be comprehended and subjectively evaluated[1]. At the root of the hierarchy is the goal (e.g. suitable car) or objectives of the problem in question, the leaf nodes are the alternatives (e.g. Mercedes, VW) which are to be compared and between these two levels are various criteria (c) and sub-criteria (sc) (e.g. c-car comfort: sc-air condition, sc-leather seat; c-car security: sc-ABS, sc-airbag and c-car body design)
STEP 3 - data collection; data is collected from domain experts corresponding to the hierarchical structure in the pairwise comparison of the alternatives on a qualitative scale. This step assesses the characteristics of each alternative (e.g. Alternative 1 (Mercedes) is much better then Alternative 2 (VW) w.r.t leather seats, airbag and car body design but Alternative 2 (VW) is better then Alternative 1 (Mercedes) considering ABS and air condition).
STEP 4 - build a pairwise comparison: for each level of criteria (sub-criteria and criteria) a pairwise comparison between the sibling nodes is to be built and organized into square matrix[4] (e.g. car security is much more important than car body design and more important than car comfort while car comfort is only a little bit more important than car body design).
STEP 5 - calculate the final result: the ratings of each alternative (cf. step 3) is multiplied by the weight of the sub-criteria (cf. step 4) and aggregated to get local ratings with respect to each criterion. The local ratings are then multiplied by the weights of the criteria (cf. step 4) and aggregated to the global ratings. The final value is used to make a decision about the problem defined in the step 1.

4 Applying AHP for the Matcher Selection

To allow a selection of matching approaches based on a mathematically rigorous method that provide a proven process for prioritization and decision-making the abovementioned process AHP is to be applied. By reducing complex decisions, i.e. which matching is suitable for a given set of requirements, to a series of pair-wise comparisons (dimensions, factors and attributes) and synthesizing the results (list of possible algorithms) decision-makers arrive at the best decision (the best matching approach) based on a clear rationale[27]. In the following we give a brief overview of how the AHP steps described in the Section 3 can be applied to the process of matcher selection taking into account some tool support for the data collection and calculation of the best alternative.
STEP 1: The problem to be solved: “Which matching approach is currently relevant w.r.t the given application requirements?”

---

[4] For details see[27]

STEP 2: The hierarchy of decision is built using the hierarchical tree described in Section 2 whereby the goal is to "find a suitable approach" (level 0) which is connected though three levels of criteria: 1^{st}level - dimensions, 2^{nd} level - factors and 3^{rd} level - attributes with the alternative matching approaches (cf. Fig. 2)

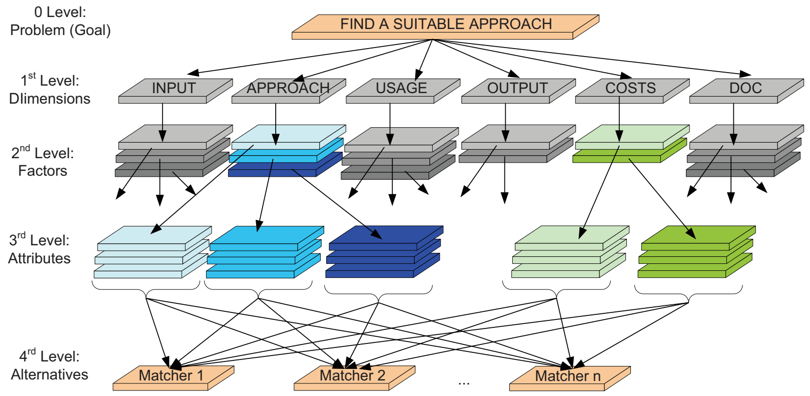


Fig. 2. AHP hierarchy structure (Detection of the suitable matching approach)

STEP 3: In order to collect data about the different alternatives of matching approaches and to be able to conduct the pairwise comparisons we firstly need the relevant information about the particular alternatives. For this reason we have developed (following the hierarchical structure of the matching characteristic) an online questionnaire (to be fill out by the domain and matching experts) that allows the addition and rating (by usage of a predefined scale from 0 to 8) of new matching alternatives. When a new matcher is added via the questionnaire into the collection of the alternatives, all available alternatives in the system are automatically weighted against the new approach. Given two matcher alternatives m_1, m_2 and criteria c as well as the user defined weighings for the single approach $w(c)_{m_1}$ and $w(c)_{m_2}$ the weighings for the pairwise comparisons (between alternatives m_1, m_2) $w(c)_{m_1,m_2}$ and $w(c)_{m_2,m_1}$ are calculated as follows: (i)$w(c)_{m_1,m_2} = w(c)_{m_1} - w(c)_{m_2}$; (ii)$w(c)_{m_2,m_1} = w(c)_{m_2} - w(c)_{m_1}$. PHPSurveyor[5] is used as the tool for providing the online questionnaire. The collected data regarding the matcher alternatives from the questionnaire is stored in a questionnaire database (MySQL) while an additional database (AHP database) stores the weighting results of the pairwise comparisons (cf. Fig. 3).

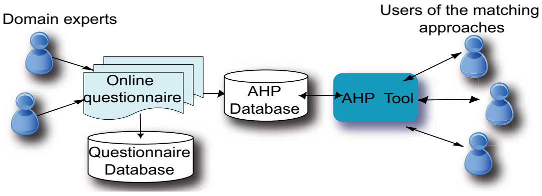
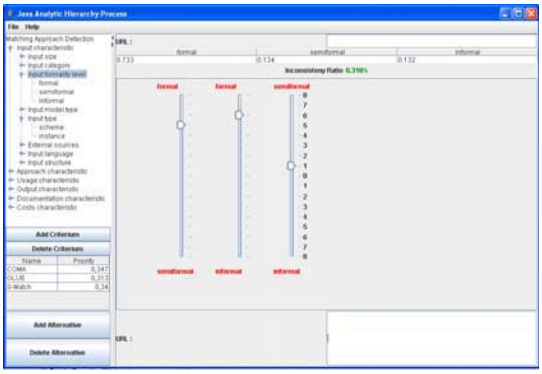


Fig. 3. AHP Tool with online questionnaire

---

[5] http://www.phpsurveyor.org

STEP 4: To enable a user-friendly pairwise comparison of the criteria from the multilevel hierarchy matcher characteristic we developed a tool which supports the processing of the AHP method[6]. Since the users of the AHP-tool have defined the requirements of their application w.r.t the suitable matching approach, they are able to weight the criteria (dimensions, factors and attributes) in the pairwise comparison on the scale from 0 - equal (two criteria have the same importance) to 8 - extremely important (one criteria is much more relevant than the other) concerning their system specification. This means, that for each level of criteria the users build a pairwise comparison between the sibling nodes: they weight the attributes against attributes, factors against factors and dimensions against dimensions (e.g. within the factor *formality level* the attribute *formal* (ontologies) is more important than *informal*, cf. Fig. 4).



Name	Priority
COMA	0,347
GLUE	0,313
S-Match	0,34

Fig. 4. AHP tool: weighed attributes

STEP 5: The decision regarding the determination of the suitable matching approach defined in the step 1 is based on the ranking $r(goal)$ of a matcher alternative m. The ranking reflects the global importance of the approach according to the alternative weightings performed in step 3 as well as criteria weightings from step 4 and is calculated as followed:

$$c_{crit} = \{n | n \text{ child of } crit\}$$

$$r(crit) = \begin{cases} |\text{getWeight}(m, crit)|, & \text{if crit is at lowest hierarchy level} \\ \sum\limits_{n \in c_{crit}} r(n) \cdot |\text{getWeight}(m, n)|, & \text{otherwise} \end{cases}$$

The higher a matcher alterative m is weighted for various criteria, with each criteria weighted with respect to the users requirements, the higher the priority of the particular approach in the entire ranking. Following this weighting process the AHP tool supports the creation of a ranking of the alternatives in depending upon the multilevel hierarchy matcher characteristic, weightings of these characteristics as well as weightings of the alternatives that shows the priority of each alternative for the defined goal.

---

[6] AHP tool is a modification of the Java AHP tool JAHP; `http://www2.lifl.fr/~morge/software/JAHP.html`

5 Conclusion

In this paper we presented the adaption of the Analytic Hierarchy Process (AHP) to the process of detection of a suitable matching approach. The proposed strategy for the decision making based on *multilevel characteristic for matching approaches* and supported by the AHP tool enables e.g. domain experts with poor expertise in ontology matching field to find appropriate approach w.r.t their application requirements. The future work will be dedicated to the collection of further matcher alternatives (with help of the online questionnaire) and the application of the AHP tool into the various Semantic Web scenarios connected with the evaluation of the entire framework.

Acknowledgements: This work has been partially supported by the Knowledge Nets project, which is part of the InterVal- Berlin Research Centre for the Internet Economy, funded by the German Ministry of Research (BMBF) and by the EU Network of Excellence KnowledgeWeb (FP6-507482).

References

1. N. Bhushan and K. Rai, editors. *Strategic Decision Making: Applying the Analytic Hierarchy Process.* Springer, 2004.
2. C. Bizer, R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, and R. Eckstein. The Impact of Semantic Web Technologies on Job Recruitment Processes. In *Proc. of the 7th Internationale Tagung Wirtschaftsinformatik 2005*, pages 1367–1383, 2005.
3. S. Castano, A. Ferrara, and S. Montanelli. Methods and Techniques for Ontology-based Semantic Interoperability in Networked Enterprise Contexts. In *Proc. of the 1st CAiSE INTEROP Workshop On Enterprise Modelling and Ontologies for Interoperability (EMOI - INTEROP 2004)*, pages 261–264, June 2004.
4. P. Clements, F. Bachman, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford, editors. *Documenting Software Architectures: Views and Beyond.* Addison-Wesley Professional, 2002.
5. H. H. Do, S. Melnik, and E Rahm. Comparison of Schema Matching Evaluations. In *Proc. of GI-Workshop "Web and Databases"*, 2002.
6. H. H. Do and E. Rahm. COMA—a system for flexible combination of schema matching approaches. In *Proc. of the 28th VLDB Conference*, 2002.
7. A. Doan, P. Domingos, and A. Halevy. Reconciling Schemas of disparate Data sources: A Machine Learning Approach. In *Proc. of the the SIGMOD01*, 2001.
8. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology Matching: A Machine Learning Approach. *Handbook on Ontologies*, pages 385–516, 2004.
9. A. Dogac, G. Laleci, Y. Kabak, and I. Cingil. Exploitingweb service semantics: Taxonomies vs. ontologies. *IEEE DATA ENGINEERING BULLETIN*, 4, 2002.
10. D. Dou, D. McDermott, and P. Qi. Ontology Translation on the Semantic Web. In *Proc. of the AInt'l Conf. on Ontologies, Databases and Applications of Semantics (ODBASE2003)*, pages 952–969, 2003.
11. N. Fenton and L. Pfleeger, editors. *Software Metrics, A Rigorous & Practical Approach.* International Thomson Cmputer Press, 1996.
12. F. Frst and F. Trichet. Axiom-based ontology matching. In *Proc. of the 3rd international conference on Knowledge capture (K-CAP'05)*, pages 195–196, New York, NY, USA, 2005. ACM Press.

13. J. Garbers, M. Niemann, and M. Mochol. A personalized hotel selection engine. In *Proc. of the Poster Session of 3rd ESWC 2006*, 2006.
14. F. Giuchiglia and P. Shvaiko. Semantic Matching. *Knowledge Web Review Journal*, pages 265–280, 2004.
15. J. R. Grandzol. Improving the faculty selection process in highe education: A case for the analytic hierarchy process. *Using Advenced Tools, Tehciques, and Methodlogies. Association for Institutional Research*, 6, 2005.
16. T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, 1995.
17. E. D. Hahn. Better decisions come from a results-based approach. *Marketing News*, 36(36):22–24, 2002.
18. W. S. Humphrey, editor. *Introduction to the Team Software Process.* Addison-Wesley Professional, 1999.
19. L. Li, B. Wu, and Y. Yang. Agent-based Ontology Integration for Ontology-based Applications. In *Proc. of the Australasian Ontology Workshop (AOW 2005)*, volume 58, pages 53–59, 2005.
20. A. Lozano-Tello and A. Gomez-Perez. ONTOMETRIC: A Method to Choose the Appropriate Ontology. *Journal of Database Management*, 15:1–18, 2004.
21. J. Madhavan, P. A. Bernstein, and E. Rham. Generic Schema Matching with Cupid. In *PROC. of the 27th VLDB Conference*, 2001.
22. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In *Proc. of the 18th International Conference on Data Engineering (ICDE02)*, 2002.
23. M. Mochol and E. Paslaru B. Simperl. Practical guidelines for building semantic erecruitment applications. In *Proc. of the International Conference on Knowledge Management (iKnow'06), Special Track: Advanced Semantic Technologies*, 2006.
24. E. Paslaru Bontas and M. Mochol. Towards a reuse-oriented methodology for ontology engineering. In *Proc. of 7th International Conference on Terminology and Knowledge Engineering (TKE 2005)*, 2005.
25. E. Paslaru Bontas, M. Mochol, and R. Tolksdorf. Case Studies on Ontology Reuse. In *Proc. of the 5th International Conference on Knowledge Management*, 2005.
26. E. Rham and P. A. Bernstein. A survey of approaches to automatic schema matching. *Journal of Very Large Data Bases*, 2001.
27. T. L. Saatly. How to Make a Decision: The Analytic Hierarchy Process. *European Journal of Operational Research*, (48):9–26, 1990.
28. P. Shvaiko. A Classification of Schema-Based Matching Approaches. Technical Report DIT-04-09, University of Trento, `http://eprints.biblio.unitn.it/archive/00000654/01/093.pdf`, December 2004.
29. P. Shvaiko. Iterative schema-besed semantic matching. Technical Report DIT-04-020, University of Trento, `http://eprints.biblio.unitn.it/archive/00000550/01/020.pdf`, June 2004.
30. P. Shvaiko and J. Euzenat. A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics*, 4:146–171, 2005.
31. G. Stumme and M. Alexander. FCA-MERGE: Bottom-up merging of ontologies. In *Proc. of the 17th IJCAI 2001*, pages 225–230, 2001.
32. M. Uschold and M. Grninger. ONTOLOGIES: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2), 1996.
33. M. Uschold and R. Jasper. A Framework for Understanding and Classifying Ontology Applications, 1999.
34. C.J. van Rijsbergen, editor. *Information retrieval, 2nd edition.* ButterworthsLondon, 1979.

Arguing over ontology alignments

L. Laera[1], V. Tamma[1], J. Euzenat[2], T. Bench-Capon[1], and T. Payne[3]

[1] Department of Computer Science, University of Liverpool,UK
[2] INRIA Rhône-Alpes, Montbonnot, France
[3] Computer Science, University of Southampton, UK

Abstract. In open and dynamic environments, agents will usually differ in the domain ontologies they commit to and their perception of the world. The availability of Alignment Services, that are able to provide correspondences between two ontologies, is only a partial solution to achieving interoperability between agents, because any given candidate set of alignments is only suitable in certain contexts. For a given context, different agents might have different and inconsistent perspectives that reflect their differing interests and preferences on the acceptability of candidate mappings, each of which may be rationally acceptable. In this paper we introduce an argumentation-based negotiation framework over the terminology they use in order to communicate. This argumentation framework relies on a formal argument manipulation schema and on an encoding of the agents preferences between particular kinds of arguments. The former does not vary between agents, whereas the latter depends on the interests of each agent. Thus, this approach distinguishes clearly between the alignment rationales valid for all agents and those specific to a particular agent.

1 Introduction

Traditionally ontologies have been used to achieve semantic interoperability between software applications, as such applications provide the definitions of the vocabularies they use to describe the world [12], and they have proved especially effective when systems are embedded in open, dynamic environments, such as the Web and the Semantic Web [4]. Interoperability relies on the ability to reconcile the differences between heterogeneous ontologies [18]. This reconciliation usually relies on the existence of correspondences (or mappings) between different ontologies (*ontology alignment* [11]), and uses them in order to interpret or translate messages exchanged by applications. Such correspondences may be generated by a variety of different matching algorithms [16] [4], and their production usually requires several steps. These can include the definition of an initial alignment, or the training over some examples, and these invariably involve some form of interpretation of preliminary results [10]. Therefore, approaches to ontology alignment can only be effective when used to support semantic interoperation at *design time* in closed or partially open environments, where the actors involved are often known, where ontology changes are controlled and thus the alignments can be established before the systems interact. However, these approaches are not sufficient to support semantic interoperation in open environments, where systems can dynamically join or leave and no prior assumption can be made on the ontologies to align. In

---

[4] A comprehensive review can be found at `http://www.ontologymatching.org`

such environments, the different systems involved need to agree on the semantics of the terms used during the interoperation, and reaching this agreement can only come through some sort of negotiation process [1].

This paper extends the notion of reaching agreement through automated negotiation (*i.e.* without human intervention) by considering the type of systems that need to interoperate, which can affect how the negotiation should proceed. Specifically, *autonomous agents* (within an open environment) may perform different tasks depending on their state and the service providers they interact with. Thus, such agents will differ in the domain ontologies they commit to [12]; and their perception of the world (and hence the choice of vocabulary used to represent concepts). Imposing a single, universally shared ontology on agents is not only impractical because it would result in assuming a standard communication vocabulary (and thus violate the dynamics of open environments) but it also does not take into account the conceptual requirements of services that could appear in future. Instead, every agent assumes its own heterogeneous private ontology, which may not be understandable by other agents. The availability of *Alignment Services* that are able to provide correspondences between two ontologies is only the beginning of a solution to achieving interoperability between agents, as any given candidate set of alignments is only suitable in certain contexts. For a given context, agents might have different and inconsistent perspectives; *i.e. interests* and *preferences*, on the acceptability of a candidate mapping, each of which may be rationally acceptable. This may be due to the subjective nature of ontologies, to the context and the requirement of the alignments and so on. For example, an agent may be interested in accepting only those mappings that have linguistic similarities, since its ontology is too *structurally simple* to realise any other type of mismatch. In addition, any decision on the acceptability of these mappings has to be made dynamically (at run time), due to the fact that the agents have no prior knowledge of either the existence or constraints of other agents.

In order to address this problem, we present a framework to support agents to negotiate agreement on the terminology they use in order to communicate, by allowing them to express their preferred choices over candidate correspondences. This is achieved by adapting argument-based negotiation to deal specifically with arguments that support or oppose the proposed correspondences between ontologies. The set of potential arguments are clearly identified and grounded on the underlying ontology languages, and the kinds of mapping that can be supported by any such argument are clearly specified. Specifically, we use a value-based argumentation framework [2], allowing each agent to express its preferences between the categories of arguments that are clearly identified in the context of ontology alignment. Our approach is able to give a formal motivation for the selection of any correspondence, and enables consideration of an agents' interests and preferences that may influence the selection of a given correspondence. Therefore, this work provides a concrete instantiation of the "meaning negotiation" process that we would like agents to achieve. Moreover, in contrast to current ontology matching procedures, the choice of alignment is based on two clearly identified elements: (i) the argumentation framework, which is common to all agents, and (ii) the preference relations which are private to each agent.

The remainder of this paper is structured as follows. Section 2 presents the argumentation framework and how it can be used. Section 3 defines the various categories of arguments that can support or attack mappings. Section 4 describes our agent model and discusses how agents should reach agreement. An example illustrating the argumentation process is given in Section 5, followed concluding remarks in Section 6[5].

2 Argumentation Framework

This paper focuses on autonomous agents situated within an open system. Each agent has a knowledge base, expressed using one of several possible ontologies. The *mental attitudes* of an agent towards correspondences are represented in terms of *interests* and *preferences*, which represent the motivations of the agent, and thus determine whether a mapping is accepted or rejected. The preferences are represented as a (partial or total) pre-ordering of preferences over different types of ontology mismatches (*Pref*)[6] .

For agents to communicate, they first need to establish a mutually acceptable set of alignments between their ontologies. Potential alignments are generated at design time (by a variety of different ontology-matching approaches [16]), and provided at run-time by a dedicated agent, called an *Ontology Alignment Service (OAS)* (Figure 1). An alignment consists of a set of correspondences between the two ontologies. A correspondence (or a mapping) can be described as a tuple: $m = \langle e, e', n, R \rangle$, where e and e' are the entities (concepts, relations or individuals) between which a relation is asserted by the correspondence; n is a degree of confidence in that correspondence; and R is the relation (e.g., equivalence, more general, etc.) holding between e and e' asserted by the correspondence [16]. A *candidate mapping* is a correspondence (provided by an OAS) that could be used by the agents to align their ontologies. Each correspondence m is accompanied by a set of justifications G, which provide an explanation as to why the correspondence was generated[7]. This information is used by the agents when generating and exchanging arguments, for and against a candidate mapping. In addition, every agent has a private threshold value ε which will be compared to the degree of confidence, n, of a mapping, to decide whether it should be considered.

In order for the agents to consider potential mappings and the reasons for and against accepting them, we use an argumentation framework based on *Value-based Argument Frameworks (VAFs)* [2], that extends Dong's classical argument system [7][8].

Definition 1. *An Argumentation Framework (AF) is a pair $AF = \langle AR, A \rangle$, where AR is a set of arguments and $A \subset AR \times AR$ is the attack relationship for AF. A comprises a set of ordered pairs of distinct arguments in AR. A pair $\langle x, y \rangle$ is referred to as "x attacks y". We also say that a set of arguments S attacks an argument y if y is attacked by an argument in S.*

---

[5] A survey of related work is given in an extended version of this paper [13].

[6] Although the agents' ontologies may differ, we eliminate the problem of integrating different ontology languages by assuming that ontologies are encoded in the same language, i.e. OWL.

[7] Although few approaches for ontology alignment provide justifications [17, 5], tools such as [9] combine different similarity metrics which can be used to provide necessary justifications.

[8] More details can be found in an extended version of this paper [13].

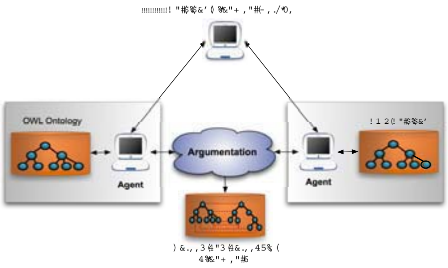


Fig. 1. Reaching agreement over ontology alignments

An argumentation framework can be simply represented as a directed graph whose vertices are the arguments and whose edges correspond to the elements of A. In this paper, we are concerned only with arguments about mappings. We can therefore define arguments as follows:

Definition 2. *An argument $x \in AF$ is a triple $x = \langle G, m, \sigma \rangle$ where m is a correspondence $\langle e, e', n, R \rangle$; G is the grounds justifying a prima facie belief that the correspondence does, or does not hold; σ is one of $\{+, -\}$ depending on whether the argument is that m does or does not hold.*

An argument x is attacked by the assertion of its negation $\neg x$, namely the *counter-argument*, defined as follows:

Definition 3. *An argument $y \in AF$ rebuts an argument $x \in AF$ if x and y are arguments for the same mapping but with different signs, e.g. if x and y are in the form $x = \langle G_1, m, + \rangle$ and $y = \langle G_2, m, - \rangle$, x counter-argues y and vice-versa.*

Moreover, if an argument x supports an argument y, they form the argument ($x \rightarrow y$) that attacks an argument $\neg y$ and is attacked by argument $\neg x$.

When the set of such arguments and counter arguments have been produced, it is necessary for the agents to consider which of them they should accept.

Definition 4. *Let $\langle AR, A \rangle$ be an argumentation framework. Let R, S, subsets of AR. An argument $s \in S$ is attacked by R if there is some $r \in R$ such that $\langle r, s \rangle \in A$. An argument $x \in AR$ is acceptable with respect to S if for every $y \in AR$ that attacks x there is some $z \in S$ that attacks y. S is conflict free if no argument in S is attacked by any other argument in S. A conflict free set S is admissible if every argument in S is acceptable with respect to S. S is a preferred extension if it is a maximal (with respect to set inclusion) admissible subset of AR.*

In addition, an argument x is *credulously accepted* if there is *some* preferred extension containing it; whereas x is *sceptically accepted* if it is a member of *every* preferred extension. The key notion here is the *preferred extension* which represents a consistent position within AF, which is defensible against all attacks and which cannot be further extended without becoming inconsistent or open to attack.

In order to take into account that, for a given situation, agents might have different point of view, we are concerned by a set of audiences, which adhere to different argument with a different strengths. Therefore we use a Value-based Argumentation Framework , which prescribes different strengths to arguments on the basis of the values they promote and the ranking given to these values by the audience for the argument. This allows us to systematically relate strengths of arguments to their motivations, and to accommodate different audiences with different interests and preferences.

Definition 5. *A* Value-Based Argumentation Framework *(VAF) is defined as $\langle AR, A, \mathcal{V}, \eta \rangle$, where (AR, A) is an argumentation framework, $\mathcal{V}$ is a set of k* values *which represent the types of arguments and η: $AR \to \mathcal{V}$ is a mapping that associates a value $\eta(x) \in \mathcal{V}$ with each argument $x \in AR$*

In section 3, the set of values $\mathcal{V}$ will be defined as the different types of ontology mismatch, which we use to define the categories of arguments and to assign to each argument one category.

Definition 6. *An* audience *for a VAF is a binary relation $\mathcal{R} \subset \mathcal{V} \times \mathcal{V}$ whose (irreflexive) transitive closure, $\mathcal{R}^*$, is asymmetric, i.e. at most one of (v, v'), (v', v) are members of $\mathcal{R}^*$ for any distinct v, $v' \in \mathcal{V}$. We say that v_i* is preferred to *v_j* in the audience *$\mathcal{R}$, denoted $v_i \succ_{\mathcal{R}} v_j$, if $(v_i, v_j) \in \mathcal{R}^*$.*

Let $\mathcal{R}$ be an audience, α is a specific audience *(compatible with $\mathcal{R}$) if α is a* total ordering *of $\mathcal{V}$* and $\forall\ v,\ v' \in \mathcal{V},\ (v, v') \in \alpha \Rightarrow (v', v) \notin \mathcal{R}^*$

In this way, we take into account that different agents (represented by different audiences) can have different perspectives on the same candidate mapping. Acceptability of an argument is defined in the following way: [9]

Definition 7. *Let $\langle AR, A, \mathcal{V}, \eta \rangle$ be a VAF and $\mathcal{R}$ an audience.*

a. For arguments x, y in AR, x is a successful attack *on y (or x* defeats *y)* with respect to the audience *$\mathcal{R}$ if:* $(x, y) \in \mathcal{A}$ and *it is* not *the case that $\eta(y) \succ_{\mathcal{R}} \eta(x)$.*

b. An argument x is acceptable to the subset *S with respect to an audience $\mathcal{R}$ if: for every $y \in AR$ that* successfully attacks *x with respect to $\mathcal{R}$, there is some $z \in S$ that successfully attacks y with respect to $\mathcal{R}$.*

c. A subset S of AR is conflict-free with respect to the audience *$\mathcal{R}$ if: for each $(x, y) \in S \times S$, either $(x, y) \notin \mathcal{A}$ or $\eta(y) \succ_{\mathcal{R}} \eta(x)$.*

d. A subset S of AR is admissible *with respect to the audience $\mathcal{R}$ if: S is conflict free with respect to $\mathcal{R}$ and every $x \in S$ is acceptable to S with respect to $\mathcal{R}$.*

e. A subset S is a preferred extension *for the audience $\mathcal{R}$ if it is a maximal admissible set with respect to $\mathcal{R}$.*

f. A subset S is a stable extension *for the audience $\mathcal{R}$ if S is admissible with respect to $\mathcal{R}$ and for all $y \notin S$ there is some $x \in S$ which successfully attacks y with respect to $\mathcal{R}$.*

In order to determine whether the dispute is resolvable, and if it is, to determine the preferred extension with respect to a value ordering promoted by distinct audiences, [2] introduces the notion of objective and subjective acceptance as follows:

---

[9] Note that all these notions are now relative to some audience.

Definition 8. *Given a VAF, $\langle AR, A, \mathcal{V}, \eta \rangle$, an argument $x \in AR$ is* subjectively acceptable *if and only if, x appears in the preferred extension for some specific audiences but not all. An argument $x \in AR$ is* objectively acceptable *if and only if, x appears in the preferred extension for* every *specific audience. An argument which is neither objectively nor subjectively acceptable is said to be* indefensible.

Next, we define the various types of arguments that can be distinguished for supporting or attacking correspondences.

3 Arguments for Correspondences

Potential arguments are clearly identified and grounded on the underlying ontology language OWL. Therefore, the grounds justifying correspondences can be extracted from the knowledge in ontologies[10]. Our classification of the grounds justifying correspondences is the following:

semantic (M): the sets of models of two entities do or do not compare;
internal structural (IS): two entities share more or less internal structure (e.g., the value range or cardinality of their attributes);
external structural (ES): the set of relations, each of two entities have, with other entities do or do not compare;
terminological (T): the names of two entities share more or less lexical features;
extensional (E): the known extension of two entities do or do not compare.

These categories correspond to the type of categorizations underlying ontology matching algorithms [18]. In our framework, we will use the types of arguments described above as types for the VAF; hence $\mathcal{V} = \{M, IS, ES, T, E\}$. For example, an audience may specify that terminological arguments are preferred to semantic arguments, or vice versa. Note that this may vary according to the nature of the ontologies being aligned. Semantic arguments will be given more weight in a fully axiomatised ontology, compared to that in a lightweight ontology where there is very little reliable semantic information on which to base such arguments.

Table 1 presents a sample set of argument schemes, instantiations of which will comprise AR. Attacks between these arguments will arise when we have arguments for the same mapping but with conflicting values of σ, thus yielding attacks that can be considered symmetric. Moreover, the relations in the mappings can also give rise to attacks: if relations are not deemed exclusive, an argument against inclusion is a fortiori an argument against equivalence (which is more general).

Example 1. Consider a candidate mapping $m = \langle c, c', \_, \equiv \rangle$ between two OWL ontologies O_1 and O_2, with concepts c and c' respectively. An argument for accepting the mapping m may be that the labels of c and c' are synonymous. An argument against may be that some of their super-concepts are not mapped.

In $VAFs$, arguments against or in favour of a candidate mapping are seen as grounded on their type. In this way, we are able to motivate the choice between preferred extensions by reference to the type ordering of the audience concerned. Moreover, the

---

[10] This knowledge includes both the extensional and intensional OWL ontology definitions.

pre-ordering of preferences *Pref* for each agent will be over $\mathcal{V}$, that corresponds to the determination of an audience.

Table 1. Argument scheme for OWL ontological alignments

Mapping	σ	**Grounds**	**Comment**
$\langle e, e', n, \equiv \rangle$	+	$\exists m_i = \langle ES(e), ES(e'), n', \equiv \rangle$	e and e' have mapped neighbours (e.g., super-entities, sibling-entities, etc.) of e are mapped in those of e'
$\langle e, e', n, \sqsubseteq \rangle$	+	$\exists m_i = \langle ES(e), ES(e'), n', \equiv \rangle$	(some or all) Neighbours (e.g., super-entities, sibling-entities, etc.) of e are mapped in those of e'
$\langle c, c', n, \sqsubseteq \rangle$	+	$\exists m_i = \langle IS(c), IS(c'), n', \equiv \rangle$	(some or all) Properties of concept c are mapped to those of concept c'
$\langle c, c', n, \sqsubseteq \rangle$	-	$\nexists m_i = \langle IS(c), IS(c'), n', \equiv \rangle$	No properties of c are mapped to those of c'
$\langle e, e', n, \equiv \rangle$	+	$\exists m_i = \langle E(e), E(e'), n', \equiv \rangle$	(some or all) Instances of e and e' are mapped
$\langle e, e', n, \sqsubseteq \rangle$	+	$\exists m_i = \langle E(e), E(e'), n', \equiv \rangle$	(some or all) Instances of e are mapped to those of e'
$\langle e, e', n, \equiv \rangle$ $\langle e, e', n, \sqsubseteq \rangle$	+	$label(e) \approx_T label(e')$	Entities's labels share lexical features (e.g., synonyms and lexical variants)
$\langle e, e', n, \equiv \rangle$ $\langle e, e', n, \sqsubseteq \rangle$	-	$label(e) \not\approx_T label(e')$	Entities' labels do not share lexical features (e.g., homonyms)

Although in $VAFs$ there is always a unique non-empty preferred extension with respect to a specific audience, provided the AF does not contain any cycles in a single argument type, an agent may have multiple preferred extensions either because no preference between two values in a cycle has been expressed, or because a cycle in a single value exists. The first may be eliminated by committing to a specific audience, but the second cannot be eliminated in this way. In our domain, where many attacks are symmetric, two cycles will be frequent and in general an audience may have multiple preferred extensions.

Thus, given a set of arguments justifying mappings organised into an argumentation framework, an agent will be able to determine which mappings are acceptable by computing the preferred extensions with respect to its preferences. If there are multiple preferred extensions, the agent must commit to the arguments present in all preferred extensions, but it has some freedom of choice with respect to those in some but not all of them.

Based on the above considerations, we thus define an *agreed correspondence* and an *agreeable correspondence* as follows. An *agreed correspondence* is the correspondence supported[11] by those arguments which are in every preferred extension of every agent. An *agreeable correspondence* is the correspondence supported by arguments which are in some preferred extension of every agent. Thus, the agents will reach a common consensus over a specific mapping m only if the mapping m is an *agreed correspondence*. However, if a mapping m is an *agreeable correspondence* for a given agent Ag, this mean that such mapping can only be considered valid and consensual for that agent.

In the next section, we present a model of agents which put forward arguments and take into account other arguments coming from their interlocutors.

4 Model of Persuasive Agents

In this paper, we are assuming a multi-agent setting containing persuasive agents that do not use the same ontology. Each agent considers the repertoire of argument schemes

---

[11] Note that a correspondence m is *supported* by an argument x if x is $\langle G, m, + \rangle$

available to it, and is able to generate a set of arguments and counter-arguments by instantiating these schemes with respect to its interests. Moreover, the agents can record their interlocutors arguments in a commitment store CS [14] and individually evaluate them. Therefore, our persuasive agent can be defined as follows:

Definition 9. *An agent Ag_i is defined by a 5-tuple $\langle O_i, VAF_i, CS_j^i, Pref, \varepsilon\rangle$ where O_i is the private ontology; $VAF_i = \langle AR_i, A_i, \mathcal{V}, \eta\rangle$ is the Valued-based Argumentation Framework of the agent Ag_i; CS_j^i is a commitment store, i.e. a set of arguments where $CS_j^i(t)$ contains propositional commitments taken before or at time t between the Ag_i and other interlocutors; $Pref$ is the private pre-ordering of preferences over $\mathcal{V}$ and ε is the private threshold value.*

The set of arguments are not necessarily disjoint. The set of arguments shared by all agents are called *common arguments*: $AR_c \subseteq \bigcap_{x \in AR_i} AR_i \in VAF_i$. Instead, the values $\mathcal{V} = \{M, IS, ES, T, E\}$ are common and shared by all audiences.

In order to take into account the arguments notified in the commitment stores, we extend the definition of valued-based argumentation framework with the following:

Definition 10. *An* extended Value-Based Argumentation Framework VAF^+ *is defined as $\langle AR^+, A^+, \mathcal{V}, \eta^+\rangle$, where $AR^+ = AR \cup \{\bigcup_{j \neq i} CS_j^i\}$. The definition of A^+ and η^+ are now related to AR^+*

Now, we can define the notion of *conviction* as follows:

Definition 11. *Let Ag_i be an agent associated with the extended valued-based argumentation framework, VAF^+ and x be an argument provided by another agent Ag_j. The agent Ag_i is convinced by the argument x iff x is acceptable with respect to all audience $\mathcal{R}$, with $Ag_i \in \mathcal{R}$.*

Given this model, in order to determine the acceptability of a potential correspondence, it needs to proceed by means of a dialectical exchange, in which a mapping is proposed, challenged and defended. Many argument protocols have been proposed, e.g. [15]. Particular dialogue games have been proposed based on Dung's Argumentation Frameworks, e.g. [8], and on VAFs [3].

In this paper, we are not considering any specific protocol or persuasive dialogue. However, the idea of a dialogue is that agents reply to each other in order to reach the interaction goal, i.e. an agreement. Thus, given a set of social and autonomous agents, and a set of potential correspondences $\{m_1, \ldots, m_i, \ldots\}$, an agent initiates a persuasion dialogue when it wants present its viewpoint to the other agents. Specifically, for each mapping m_i, if the agent wants to accept that mapping, it will put forward arguments for m_i. In the negative case, it will put forward arguments against. If the other agents have no arguments against/for the mapping, it closes the dialogue. If the players have the same convictions, the the arguments is acceped and the dialogue closes. Otherwise, the goal of the dialogue is the resolution of the conflict by verbal means, and thus with an exchange of arguments and counter-arguments.

The dialogue between agents can thus consist simply of the exchange of individual arguments, from which they can compute acceptable mappings over the CS, by computing the preferred extensions. If necessary and desirable, these can then be reconciled into a mutually acceptable position through a process of negotiation, as suggested in [6], which defines a dialogue process for evaluating the status of arguments in a VAF, and shows how this process can be used to identify mutually acceptable arguments.

In [13] a detailed approach to argue over alignments and complete argumentation framework, with a common set of arguments, is proposed.

5 A Walk through Example

Let us assume that some agents or services need to interact with each other using two independent but overlapping ontologies. The first agent, Ag_1 uses the bibliographic ontology[12] from the University of Toronto, based on bibTeX; whereas the second agent, Ag_2, uses the General University Ontology[13] from Mondeca[14]. For space reasons, we only consider a subset of these ontologies, shown in Table 2, where the first and second ontologies are represented by O_1 and O_2 respectively.
We will assume that the set of candidate mappings, provided by the Ontology Alignment Service (OAS), is the following::

$$m_1{=}\langle O_1\colon Press, O_2\colon Periodical, n, =\rangle;$$ [15]
$$m_2{=}\langle O_1\colon publication, O_2\colon Publication, n, =\rangle;$$
$$m_3{=}\langle O_1\colon hasPublisher, O_2\colon publishedBy, n, =\rangle;$$
$$m_4{=}\langle O_1\colon Magazine, O_2\colon Magazine, n, =\rangle;$$
$$m_5{=}\langle O_1\colon Newspaper, O_2\colon Newspaper, n, =\rangle;$$
$$m_6{=}\langle O_1\colon Organization, O_2\colon Organization, n, =\rangle.$$

The generation of the arguments and counter-arguments of the Ag_1 and Ag_2 are achieved by instantiating the argumentation schemes, discussed previously, with respect to the agent's preferences and threshold. However, here we assume a degree of confidence n that is above the threshold of both agent, and so will not influence their acceptability. Assume now that there are two possible audiences, $\mathcal{R}_1$, which prefers terminology to external structure, ($T \succ_{\mathcal{R}_1} ES$), and $\mathcal{R}_2$, which prefers external structure to terminology ($ES \succ_{\mathcal{R}_2} T$). The pre-ordering of preference *Pref* will correspond to the agent's audience. The agents Ag_1 and Ag_2 take on the part, respectively, of the audience $\mathcal{R}_1$ and $\mathcal{R}_2$. For space reasons, we will only evaluate the mapping m_1[16].

The argumentation starts, with the agent Ag_1 that wants to reject the mapping m_1 and will thus argue against it, forwarding an argument A. A states that none of the super-concepts of the concept $O_1\colon Press$ are mapped to any super-concept of $O_2\colon Periodical$. The agent Ag_2, instead, does not agree and counter-argues with an argument B. B argues for m_1, because two sub-concepts of $O_1\colon Press$, $O_1\colon Magazine$ and $O_1\colon Newspaper$, are mapped to two sub-concepts of $O_2\colon Periodical$, $O_2\colon Magazine$ and $O_2\colon Newspaper$, as established by m_4 and m_5. The agent Ag_1 attacks B with the argument C, because $O_1\colon Press$ and $O_2\colon Periodical$ do not have any lexical similarity. The agent Ag_2

---

[12] http://www.cs.toronto.edu/semanticweb/maponto/ontologies/BibTex.owl

[13] http://www.mondeca.com/owl/moses/univ.owl

[14] Note that ontology O_2 has been slightly modified for the purposes of this example.

[15] m_1 states an equivalence correspondence with confidence n between the concept $Press$ in the ontology O_1 and the concept $Periodical$ in the ontology O_2

[16] An extended version of this example is provided in [13] .

Table 2. Excerpts of O_1 and O_2 ontologies

O_1 **Ontology**	O_2 **Ontology**
$Artifact \sqsubseteq \top$	$Document \sqsubseteq \top$
$Print\_Media \sqsubseteq Artifact$	$Publication \sqsubseteq Document$
$Press \sqsubseteq Print\_Media$	$Periodical \sqsubseteq Publication$
$Magazine \sqsubseteq Press$	$Magazine \sqsubseteq Periodical$
$Newspaper \sqsubseteq Press$	$Newspaper \sqsubseteq Periodical$
$publication \sqsubseteq \forall hasPublisher.Publisher$	$Newsletter \sqsubseteq Periodical$
$publication \sqsubseteq Print\_Media$	$Journal \sqsubseteq Periodical$
$Publisher \sqsubseteq Organization$	$Publication \sqsubseteq Document$
	$Publication \sqsubseteq \forall publishedBy.Organization$

does not have any other argument to reply to C but it supports the correspondences m_4, m_5 and m_6 by six arguments. K, L and M justify the mapping m_4, since, respectively, the labels of O_1: $Magazine$ and O_2: $Magazine$ are lexically similar; their siblings are mapped, as established by m_5, and their super-concepts; O_1: $Press$ and O_2: $Periodical$ are mapped by m_1. There is a similar situation for the arguments M, N and O. Clearly, argument A attacks the arguments D and I.

This position is illustrated in Figure 2, where nodes represent arguments (labelled with their Id) with the respective type value $\mathcal{V}$. The arcs represent the attacks A, whereas the direction of the arcs represents the direction of the attack.

Table 3 shows these arguments, labelled with an identifier Id, its type $\mathcal{V}$, and the attacks A that can be made on it by opposing arguments.

Table 3. Arguments for and against the correspondences m_1, m_4 and m_5

Id	**Argument**	A	$\mathcal{V}$
A	$\langle \nexists m = \langle superconcept(Press), superconcept(Periodical), n, \equiv, \rangle, m_1, - \rangle$	B,D,I	ES
B	$\langle \exists m = \langle subconcept(Press), subconcept(Periodical), n, \equiv, \rangle, m_1, + \rangle$	A,C	ES
C	$\langle Label(Press) \not\approx_T Label(Periodical), m_1, - \rangle$	B	T
D	$\langle Label(Magazine) \approx_T Label(Magazine), m_4, + \rangle$		T
E	$\langle \exists m = \langle siblingConcept(Magazine), siblingConcept(Magazine), n, \equiv, \rangle, m_4, + \rangle$		ES
F	$\langle \exists m = \langle superconcept(Magazine), superconcept(Magazine), n, \equiv, \rangle, m_4, + \rangle$		ES
G	$\langle Label(Newspaper) \approx_T Label(Newspaper), m_5, + \rangle$		T
H	$\langle \exists m = \langle siblingConcept(Newspaper), siblingConcept(Newspaper), m_5, + \rangle$		ES
I	$\langle \exists m = \langle superconcept(Newspaper), superconcept(Newspaper), n, \equiv, \rangle, m_5, + \rangle$		ES

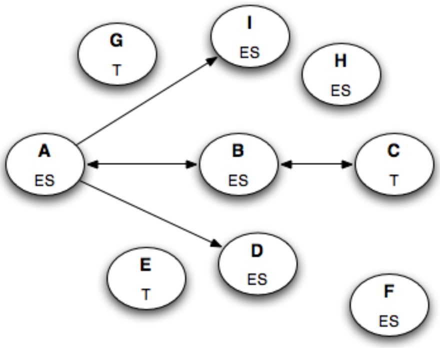


Fig. 2. Value-Based Argumentation Frameworks

Finally, we can compute the acceptability of the arguments, computing the preferred extensions (see Table 5). Therefore, the arguments accepted by both audiences are $\{D, E, G, H\}$. Arguments A, C are, however, both potentially acceptable, since both audiences can choose to accept them, as they appear in some preferred extension for each audience. This means that the mapping m_1 will be rejected for the agent Ag_1 (since B is unacceptable to $\mathcal{R}_1$), while the mappings m_4 and m_5 will both be accepted (they are both accepted by $\mathcal{R}_1$ and both acceptable to $\mathcal{R}_2$). The *agreed correspondence* are then m_4 and m_5.

Table 4. Preferred Extensions

Preferred Extensions for the framework (a)	**Audience**
$\{A, C, D, E, G, H\}$	$\mathcal{R}_1$
$\{A, C, D, E, F, G\}$, $\{B, I, D, E, F, G\}$	$\mathcal{R}_2$
$\{A, C, D, E, F, G\}$, $\{B, I, D, E, F, G\}$	

6 Summary and Outlook

In this paper we have outlined a framework that provides a novel way for agents, who use different ontologies, to argue and reach agreement over ontology alignment. This is achieved using an argumentation process in which candidate correspondences are accepted or rejected, based on the ontological knowledge and the agent's preferences. Argumentation is based on the exchange of arguments, against or in favour of a correspondence, that interact with each other using an *attack* relation. Each argument instantiates an argumentation schema, and utilises domain knowledge, extracted from extensional and intensional ontology definitions.

Our approach is able to give a formal motivation for the selection of a correspondence, and enables consideration of an agent's interests and preferences that may influence the selection of a correspondence. We believe that this approach will aim at reaching mutual understanding and communicative work in agents system more sound and effective. Future work will include experimental testing in order to demonstrate the practicality of our approach. An interesting topic for future work would be to investigate how to argue about the whole alignments, and not only the individual candidate mapping.

7 Acknowledgements

The research has been partially supported by Knowledge Web (FP6-IST 2004-507482) and PIPS (FP6-IST 2004-507019). Special thanks to Floriana Grasso and Ian Blacoe for their comments.

References

1. K. Aberer and et al. Emergent semantics principles and issues. In *Proceedings of Database Systems for Advances Applications, 9th International Conference, DASFAA 2004*, 2004.

2. T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. In *Journal of Logic and Computation*, volume 13, pages 429–448, 2003.
3. T. J. M. Bench-Capon. Agreeing to differ: Modelling persuasive dialogue between parties without a consensus about values. In *Informal Logic*, volume 22, pages 231–245, 2002.
4. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
5. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. imap. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 383–394.
6. S. Doutre, T. Bench-Capon, and P. E. Dunne. Determining preferences through argumentation. In *Proceedings of AI*IA'05*, pages 98–109, 2005.
7. P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. In *Artificial Intelligence*, volume 77, pages 321–358, 1995.
8. P. Dunne and T. J. M. Bench-Capon. Two party immediate response disputes: Properties and efficiency. In *Artificial Intelligence*, volume 149, pages 221–250, 2003.
9. M. Ehrig and S. Staab. Qom - quick ontology mapping. In *Proceedings of the International Semantic Web Conference*, 2004.
10. J. Euzenat. Alignment infrastructure for ontology mediation and other applications. In Hepp, editor, *Proceedings of the First International workshop on Mediation in semantic web services*, 2005.
11. J. Euzenat and P. Valtchev. Similarity-based ontology alignment in owl-lite. In *Proceedings of European Conference on Artificial Intelligence (ECAI 04)*, 2004.
12. T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
13. L. Laera, V. Tamma, J. Euzenat, T. Bench-Capon, and T. Payne. Reaching agreements over ontology alignments. In *Proceedings of the Fifth International Semantic Web Conference (ISWC'06)*, 2006.
14. N. Maudet and B. Chaib-draa. Commitment-based and dialogue-game based protocols-news trends in agent communication language. *Knowledge Engineering Review*, 17(2):157–179, 2003.
15. P. McBurney and S. Parsons. Locutions for argumentation in agent interaction protocols. In *Proceedings of International Workshop on Agent Communication, New-York (NY US)*, pages 209–225, 2004.
16. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on data semantics*, 4:146–171, 2005.
17. P. Shvaiko, F. Giunchiglia, P. Pinheiro da Silva, and D. McGuinness. Web explanations for semantic heterogeneity discovery. In *Proceedings of ESWC*, pages 303–317, 2005.
18. P. Visser, D. Jones, T. Bench-Capon, and M. Shave. Assessing heterogeneity by classifying ontology mismatches. In N. Guarino, editor, *Proceedings of the FOIS'98*, 1998.

Improving Automatically Created Mappings using Logical Reasoning

Christian Meilicke[1], Heiner Stuckenschmidt[1], Andrei Tamilin[2]

[1] University of Mannheim, Germany
[2]ITC-irst and University of Trento, Italy

Abstract. A lot of attention has been devoted to heuristic methods for discovering semantic mappings between ontologies. Despite impressive improvements, the mappings created by these automatic matching tools are still far from being perfect. In particular, they often contain wrong and redundant mapping rules. In this paper we present an approach for improving such mappings using logical reasoning in the context of Distributed Description Logics (DDL). Our method is orthogonal to the matching algorithm used and can therefore be used in combination with any matching tool. We explain the general idea of our approach informally using a small example and present the results of experiments conducted on the OntoFarm Benchmark which is part of the Ontology Alignment Evaluation challenge.

1 Motivation

The problem of semantic heterogeneity is becoming more and more pressing in many areas of information technologies. The Semantic Web is only one area where the problem of semantic heterogeneity has lead to intensive research on methods for semantic integration. The specific problem of semantic integration on the Semantic Web is the need to not only integrate data and schema information, but to also provide means to integrate ontologies, rich semantic models of a particular domain. There are two lines of work connected to the problem of a semantic integration of ontologies:

- The (semi-) automatic detection of semantic relations between ontologies (e.g., [9, 6, 11, 12, 7]).
- The representation and use of semantic relations for reasoning and query answering (e.g., [14, 10, 5, 3, 2]).

So far, work on representation of and reasoning with mappings has focussed on mechanisms for answering queries and using mappings to compute subsumption relationships between concepts in the mapped ontologies. These methods always assumed that the mappings used are manually created and of high quality (in particular consistent). In this paper we investigate logical reasoning about mappings that are not assumed to be perfect. In particular, our methods can be used to check (automatically created) mappings for formal and conceptual consistency and determine implied mappings that have not explicitly been represented. We investigate such mappings in the context of Distributed Description Logics [1, 13], an extension of traditional description logics with mappings between concepts in different T-boxes. The functionality described in this paper will become more important in the future because more and more ontologies are created and need to be linked. For larger ontologies the process of mapping will not be done completely by hand, but will rely on or will at least be supported by automatic mapping approaches. We see our work as a contribution to semi-automatic approaches for creating mappings between ontologies where possible mappings are computed automatically and then corrected manually making use of methods for checking the formal and conceptual properties of the mappings.

In previous work we have proposed a number of formal properties of mappings in Distributed Description Logics that we consider useful for judging the quality of a set of mappings [16]. In this paper, we refine and extend this work in several directions.

Debugging of mappings We propose a process for (semi-)automatically debugging automatically created mappings making use of some of the properties mentioned above. In particular we use the notion of mapping consistency to detect problems caused by the mappings. For each potential problem, we determine the minimal set of mapping rules responsible for the problem (minimal conflict set). For each conflict set, we try to identify which mapping rule is incorrect and remove it form the mapping.

Implementation On top of the DRAGO reasoning system [15] we built a prototype of mapping debugger for computing minimal conflict sets with respect to an inconsistency caused by a mapping as well as some heuristics for automatic repairing of an inconsistent mapping. We further added a minimization functionality for computing minimal mapping sets from redundant ones.

Experiments We tested the approach using the OntoFarm data set, a set of several rich OWL ontologies describing the domain of conference management systems [17]. We used the CtxMatch matching tool to automatically create mappings between each of the ontologies. We further automatically determined problems (in particular unsatisfiable concepts) created by the mapping and tried to fix them automatically using the debugging process proposed in this paper. In the concluding step of the experimental study, we tried to compute for each mapping its logically-equivalent minimal version.

The structure of the paper is as follows. We start with a brief recall of basic definitions of Distributed Description Logics and explanations of the reasoning mechanisms. Then we describe the intuitions of our debugging/minimization approaches using a small example. Finally, we report on some preliminary experimental evaluation of the techniques proposed in this paper and summarize the results.

2 Distributed Description Logic

Distributed Description Logic framework (DDL) is a formal tool for representing and reasoning with multiple ontologies pairwise linked by semantic mappings. In this section, we briefly recall some key definitions and properties of DDL relying on the original studies in [1, 13].

2.1 Syntax and Semantics

Given a set I of indexes, used to enumerate a set of ontologies, a *Distributed Description Logics* is then a collection $\{\mathcal{D}L_i\}_{i\in I}$ of Description Logics. Each ontology i is formalized by a T-box $\mathcal{T}_i$ of $\mathcal{D}L_i$, so that the initial set of ontologies in DDL corresponds to a family of T-boxes $\mathcal{T} = \{\mathcal{T}_i\}_{i\in I}$. To distinguish the descriptions from various $\mathcal{T}_i$ in the family, DDL utilizes a prefix notation to pin descriptions to ontologies where they are considered in, e.g., $i : X$, $i : X \sqsubseteq Y$. Semantic relations between pairs of ontologies a represented in DDL by bridge rules. A *bridge rule* from i to j is an expression of the following two forms:

$i : X \xrightarrow{\sqsubseteq} j : Y$ – an *into-bridge rule*

$i : X \xrightarrow{\sqsupseteq} j : Y$ – an *onto-bridge rule*

where X and Y are concepts of ontologies $\mathcal{T}_i$ and $\mathcal{T}_j$ respectively. The derived bridge rule $i : X \xrightarrow{\equiv} j : Y$ can be defined as the conjunction of corresponding into- and onto-bridge rule.

Intuitively, the into-bridge rule $i : Bachelor \xrightarrow{\sqsubseteq} j : Student$ states that, from the j-th point of view the concept $Bachelor$ in i is more specific than its local concept $Student$. Similarly, the onto-bridge rule $i : ScientificEvent \xrightarrow{\sqsupseteq} j : Conference$ expresses the more generality relation.

A *distributed T-box* $\mathfrak{T} = \langle \mathcal{T}, \mathfrak{B} \rangle$ consists of a collection of T-boxes $\mathcal{T} = \{\mathcal{T}_i\}_{i\in I}$ and a collection of bridge rules $\mathfrak{B} = \{\mathfrak{B}_{ij}\}_{i\neq j\in I}$ between them.

The semantics of DDL is based on the key assumption that each ontology $\mathcal{T}_i$ in the family is *locally* interpreted by interpretation $\mathcal{I}_i$ on its *local* interpretation domain $\Delta^{\mathcal{I}_i}$. The semantic correspondences between heterogeneous local domains, e.g., the representations of a registration fee in US Dollars and in Euro, are modeled in DDL by a domain relation.

A *domain relation* r_{ij} represents a possible way of mapping the elements of $\Delta^{\mathcal{I}_i}$ into the domain $\Delta^{\mathcal{I}_j}$: $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ such that r_{ij} denotes $\{d' \in \Delta^{\mathcal{I}_j} \mid \langle d, d' \rangle \in r_{ij}\}$; for any subset D of $\Delta^{\mathcal{I}_i}$, $r_{ij}(D)$ denotes $\bigcup_{d\in D} r_{ij}(d)$; and for any $R \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ $r_{ij}(R)$ denotes $\bigcup_{\langle d,d'\rangle\in R} r_{ij}(d) \times r_{ij}(d')$. For instance, if $\Delta^{\mathcal{I}_1}$ and $\Delta^{\mathcal{I}_2}$ are the representations of a registration fee in US Dollars and in Euro, then r_{12} could be a rate of exchange function, or some other approximation relation.

A *distributed interpretation* $\mathfrak{I} = \langle \{\mathcal{I}_i\}_{i\in I}, \{r_{ij}\}_{i\neq j\in I} \rangle$ of a distributed T-box $\mathfrak{T} = \langle \mathcal{T}, \mathfrak{B} \rangle$ consists of a family of local interpretations $\mathcal{I}_i$ on local interpretation domains $\Delta^{\mathcal{I}_i}$, one for each $\mathcal{T}_i$, and a family of domain relations r_{ij} between these local domains. A distributed interpretation $\mathfrak{I}$ is said to satisfy a distributed T-box $\mathfrak{T} = \langle \mathcal{T}, \mathfrak{B} \rangle$, written $\mathfrak{I} \models \mathfrak{T}$, if all T-boxes in $\mathcal{T}$ are satisfied

$$\mathfrak{I} \models \mathcal{T}_i, \text{ if } \mathcal{I}_i \models A \sqsubseteq B \text{ for all } A \sqsubseteq B \in \mathcal{T}_i$$

and all bridge rules in $\mathfrak{B}$ are satisfied:

$$\mathfrak{I} \models i : X \xrightarrow{\sqsubseteq} j : Y, \text{ if } r_{ij}(X^{\mathcal{I}_i}) \subseteq Y^{\mathcal{I}_j}$$
$$\mathfrak{I} \models i : X \xrightarrow{\sqsupseteq} j : Y, \text{ if } r_{ij}(X^{\mathcal{I}_i}) \supseteq Y^{\mathcal{I}_j}$$

Given a distributed T-box $\mathfrak{T} = \langle \mathcal{T}, \mathfrak{B} \rangle$, one can perform some basic Distributed DL inferences. A concept $i : C$ is *satisfiable* with respect to $\mathfrak{T}$ if there exist a distributed interpretation $\mathfrak{I}$ of $\mathfrak{T}$ such that $C^{\mathcal{I}_i} \neq \emptyset$. A concept $i : C$ is *subsumed* by a concept $i : D$ with respect to $\mathfrak{T}$ ($\mathfrak{T} \models i : C \sqsubseteq D$) if for every distributed interpretation $\mathfrak{I}$ of $\mathfrak{T}$ we have that $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$.

2.2 DDL Inference Mechanisms

Although both in DL and Distributed DL the fundamental reasoning services lay in verification of concepts satisfiability/subsumption within a certain ontology, in DDL, besides the ontology itself, the reasoning also depends on other ontologies that affect it through semantic mappings. This affection consist in the ability of bridge rules to *propagate the knowledge* across ontologies in form of subsumption axioms.

The simplest case illustrating the knowledge propagation in DDL is the following:

$$\frac{i : A \sqsubseteq B, \quad i : A \xrightarrow{\sqsupseteq} j : G,\ i : B \xrightarrow{\sqsubseteq} j : H}{j : G \sqsubseteq H} \tag{1}$$

In languages that support disjunction, the simplest propagation rule can be generalized to the propagation of subsumption between a concept and a disjunction of other concepts in the following way:

$$\frac{i : A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n, \quad i : A \xrightarrow{\sqsupseteq} j : G,\ i : B_k \xrightarrow{\sqsubseteq} j : H_k \ (1 \leq k \leq n)}{j : G \sqsubseteq H_1 \sqcup \ldots \sqcup H_n} \tag{2}$$

The important property of the described knowledge propagation is that it is directional, i.e., bridge rules from i to j support knowledge propagation only from i towards j. It has been shown in [13] that adding the inference pattern (2) to existing DL tableaux reasoning methods lead to a correct and complete method for reasoning in DDL. This method has been implemented in the DRAGO DDL reasoner.

3 The Debugging Process

In this section we will explain the general idea of our approach for improving automatically created mappings based on reasoning about mappings in Distributed Description Logics using a simple example. In particular, we consider two ontologies in the domain of conference management systems, the same domain we did our experiments in. For each ontology, i and j, we only consider a single axiom, namely:

$$i : Author \sqsubseteq Person \qquad \text{and} \qquad j : Person \sqsubseteq \neg Authorization$$

These simple axioms that describe the concept of a person in two different ontologies – one stating that an author is a special kind of person and the other one stating that the concepts Person and Authorization (to access submitted papers) are disjoint concept – are enough to explain the important features of our approach. The approach consists of the following steps.

3.1 Mapping Creation

In the first step, we use any existing system for matching ontologies to create an initial set of mapping hypotheses. In particular, we are interested in mappings between class names, because these are the kinds of mappings that we can reason about using DDL framework. In order to support automatical repair of inconsistent mappings later on, the matching algorithm chosen should ideally not only return a set of mappings, but also a level of confidence in the correctness of a mapping. For the sake of simplicity, we assume that we use a simple string matching method that compares the overlap in concept names and computes a similarity value that denotes the relative size of the common substring[1]. Mappings are created based on a threshold for this value that we assume to be 1/3. Applying this method to the example will result in the following two mappings with corresponding levels of confidence:

$$i : Person \xrightarrow{\equiv} j : Person, 1.00$$
$$i : Author \xrightarrow{\equiv} j : Authorization, 0.46$$

We further assume that the mapping method also applies some structural heuristics to derive additional mappings and propagates the levels of confidence accordingly. For instance, the fact that $i : Person$ is a superconcept of $i : Author$ which is assumed to be equivalent to $j : Authorization$ may be used to derive the following mapping:

$$i : Person \xrightarrow{\sqsupseteq} j : Authorization, 0.46$$

In the same way, the fact that $i : Author$ is a subconcept of $i : Person$ and the fact that $i : Person$ is assumed to be equivalent to $j : Person$ may be used to the following addition mapping:

$$i : Author \xrightarrow{\sqsubseteq} j : Person, 1.00$$

We can easily see that the process has produced two incorrect mappings, namely the ones with a confidence of 0.46. It could be argued that it is easy to get rid of these incorrect mappings by raising the threshold to 0.5 for instance. This however is no sustainable solution to the problem, because there might be mappings with a level of confidence below 0.5 that are correct, on the other hand, there might still be incorrect mappings with a confidence of more than 0.5. Instead of relying on artificially set thresholds, we propose to analyze the impact of created mappings on the connected ontologies and to eliminate mappings that have a malicious influence.

3.2 Diagnosis

The mapping set described in the last step now serves as a basis for analyzing the effect of mappings and detecting malicious mappings. This process is similar to the well known concept of model-based diagnosis which has already successfully been applied to the task of detecting wrong axioms in single ontologies. Similar to existing approaches for diagnosing ontologies, our starting point are unsatisfiable concepts which

---

[1] of course we use more sophisticated methods in the real experiments

are interpreted as symptoms for which a diagnosis has to be computed. Compared to the general task of diagnosing ontologies, we are in a lucky position, because we have to deal with a much smaller set of potential diagnosis. In particular, we claim that the ontologies connected in the first step do not contain unsatisfiable concepts. If we now observe unsatisfiable concepts in the target ontology[2] and assuming that the ontologies themselves are correct, we know that they have to be caused by some mappings in the mapping set.

To illustrate this situation, we can have a look at our example again. Using existing techniques for reasoning in DDL, we can derive that the concept Authorization is globally unsatisfiable, i.e., $j : Authorization^{\mathcal{I}} = \emptyset$, because we have $Authorzation \sqsubseteq \neg Person$ and at the same time, we can infer $Authorization \sqsubseteq Person$. There are two reasons for this, namely:

$$j : Authorization^{\mathcal{I}_j} = r_{ij}(i : Author^{\mathcal{I}_i}) \subseteq r_{ij}(i : Person^{\mathcal{I}_i}) = j : Person^{\mathcal{I}_j}$$

and

$$j : Authorization^{\mathcal{I}_j} \subseteq r_{ij}(i : Person^{\mathcal{I}_i}) = j : Person^{\mathcal{I}_j}$$

Interpreting the inconsistency of the concept $j : Authorization$ as a symptom, we can now try to identify and repair the cause of this inconsistency. For this purpose, we compute irreducible conflict set for this symptom. Here an irreducible conflict set is a set of mappings that makes the concept unsatisfiable and has the additional property that removing a mapping from the set makes the concept satisfiable again. the arguments above it is easy to see that he have the following irreducible conflict sets:

$$\{i : Person \xrightarrow{\equiv} j : Person, i : Author \xrightarrow{\equiv} j : Authorization\}$$

and

$$\{i : Person \xrightarrow{\equiv} j : Person, i : Person \xrightarrow{\sqsupseteq} j : Authorization\}$$

In classical diagnosis, all conflict sets[3] are computed and the diagnosis is computed from these conflict sets using the hitting set algorithm. For the case of diagnosing mappings this is neither computationally feasible nor does it provide the expected result. In our example, the hitting set would consist of the mapping $i : Person \xrightarrow{\equiv} j : Person$ which, as we sill see later, is the only mapping that actually carries some correct information.

Our solution to the problem is to use an iterative approach that computes an often not minimal hitting set by determining one conflict set at a time and immediately fixing it in the way described in the next section. In our example, the algorithm will first detect the second conflict and fix it, afterwards, the method checks whether the concept $j : Authorization$ is still inconsistent. As this is the case, the second conflict set will be detected and fixed as well removing the problem.

---

[2] the formal semantics of DDL guarantees that the addition of mappings cannot lead to unsatisfiable concepts in the source ontology

[3] in classical diagnosis often only minimal conflict sets are considered

3.3 Heuristic Debugging

As mentioned above, the result of the diagnosis step is an irreducible conflict sets, in particular a set of mappings that make a concept unsatisfiable and with the additional property that removing one mapping from this set solves the problem in the sense that the concept becomes satisfiable. The underlying idea of our approach is now that unsatisfiable concepts are the result of wrong mappings. This means that each irreducible conflict set contains at least one mapping rule that does state a correct semantic relation between concepts and therefore should not be in the set of mappings. The goal of the debugging step is now to identify this malicious mapping and remove it from the overall mapping set. If we chose the right mapping for removal the quality of the overall mapping set should be improved, because a wrong mapping has been removed. In the case of our example, the first irreducible conflict set that will be considered consists of the following two mappings one of which we have to remove:

$$i : Person \xrightarrow{\equiv} j : Person, 1.00$$
$$i : Author \xrightarrow{\equiv} j : Authorization, 0.46$$

There are different ways now, in which a decision about the mapping to remove could be made. The easiest way is to use an interactive approach where the conflict sets are presented to a human user who decides which mapping should be removed. In our case, the user will easily be able to decide that the mapping $i : Author \xrightarrow{\equiv} j : Authorization$ is not correct and should be removed. In the second iteration, the following two mappings will be in the irreducible conflict set:

$$i : Person \xrightarrow{\equiv} j : Person, 1.00$$
$$i : Person \xrightarrow{\sqsupseteq} j : Authorization, 0.46$$

For this set the user will be able to see immediately that the second mapping should be removed, because it is not correct. This approach sound trivial, but in the presence of large mapping sets, providing the user with feedback about potential problems in terms of small conflict sets is of great help and often reveals problems that are hard to see when looking at the complete mapping set.

We can also try to further automate the debugging process by letting the system decide, which mapping rule to eliminate. In cases where the matching system already provides a measure of confidence, this is again quite simple, as we can simply remove the mapping rule with the lowest degree of confidence. In our case this is again the rule $i : Author \xrightarrow{\equiv} j : Authorization$ and removing it will lead to a better mapping set. It is not always possible, however, to rely on the confidence provided by the matching system, either because the system simply does not provide any or because the levels of confidence provided are not informative. In our experiments, we often had the situation where all mapping even though they were conflicting had a confidence of 100% attached. In this case, we have to think of a new way of ranking mappings. An approach that we used in our experiments that turned out to work quite well is to compute the semantic distance of the concept names involved using WordNet synsets. For the example above it is clear that this heuristic will also lead to an exclusion of the second rule, because the class names in the first rule are equivalent and therefore have the least semantic distance possible. In cases where no distinction can be made using this heuristic, we have to switch back to the interactive mode and ask the user which mapping to remove. In any cases, the debugging step leaves us with a single mapping that does not create any inconsistencies. In order to get a complete set of correct mappings, we can now infer all additional mappings that follow from this one which leads us to the corrected final set of mappings in our case this final set if the following.

$$\begin{gathered} i : Person \xrightarrow{\equiv} j : Person, 1.00 \\ i : Author \xrightarrow{\sqsupseteq} j : Person, 1.00 \end{gathered}$$

In summary, the process above is a way to improve the quality of automatically generated mapping sets by means of intelligent post-processing. Using formal properties of mappings and logical reasoning we are able to detect wrong mappings by analyzing their impact and tracking unwanted effects back to the mapping rules that caused them. In this our method is not yet another ontology matching method, but it is actually orthogonal to existing developments in the area of ontology matching as it can be applied to any set of mappings. The approach can be extended in several directions. First of all we can use symptoms other than concept satisfiability as a starting point for diagnosis. Further, we can use the method on joint sets of competing mappings created by different matching algorithms. This will help us to get a better coverage of the actual semantic relations and the trust in the quality of the different matching algorithms provides us with an additional criterion for selecting mappings to be discarded.

3.4 Minimization

A further improvement of the debugged mapping can be achieved by removing redundant mappings - mappings that logically follow from other mappings. In [16] we defined the notion of minimality of a mapping that we use in this context to remove redundant mappings. In the example for instance, the two mappings derived using structural heuristics do not really add new information to the system, because they can be derived from the two equivalence mappings that have been created first. In particular $i : Person \xrightarrow{\sqsupseteq} j : Authorization$, is redundant information, because:

$$\begin{aligned} & i : Author^{\mathcal{I}_i} \subseteq i : Person^{\mathcal{I}_i} \tag{3} \\ \Longrightarrow \quad & r_{ij}(Author^{\mathcal{I}_i}) \subseteq r_{ij}(Person^{\mathcal{I}_i}) \tag{4} \\ & r_{ij}(Person^{\mathcal{I}_i}) = j : Person^{\mathcal{I}^j} \tag{5} \\ \Longrightarrow \quad & r_{ij}(Author) \subseteq j : Person^{\mathcal{I}_j} \tag{6} \end{aligned}$$

This means that for reasoning with automatically created mappings, we only have to take into account the equivalence mapping between the person concept in the two ontologies, because it is the basis for inferring the other one. For this reason, we remove all mappings that can be shown to be redundant in the sense that they can be derived from using other mappings from the set of mappings and only continue with the resulting minimal mapping set that still carries all the semantics of the complete set.

4 Experiments

In this section we report on some preliminary experimental evaluation of the mapping debugging/minimization techniques presented in the preceding sections. All the experiments have been conducted on the prototype of the debugger/minimizer implemented on top of the DRAGO DDL reasoner [15].

4.1 Experimental Setting

To perform experiments, we used a set of ontologies developed in the OntoFarm project [17] which are used as a part of Benchmark in Ontology Alignment Evaluation challenge.[4] In particular, we selected several ontologies modeling the domain of *conference organization*:

Ontology	Description Logics Expressivity	Number of classes	Number of properties
CMT	$\mathcal{ALCIF}(\mathcal{D})$	30	59
CONFTOOL	$\mathcal{SIF}(\mathcal{D})$	39	36
CRS	$\mathcal{ALCIF}(\mathcal{D})$	14	17
EKAW	$\mathcal{SHIN}$	73	33
PCS	$\mathcal{ALCIF}(\mathcal{D})$	24	38
SIGKDD	$\mathcal{ALCI}(\mathcal{D})$	51	28

Given this ontology test set, we apply the following experimental scenario. Using the CtxMatch matching tool [4], we automatically compute mappings between pairs of ontologies in the test set. Among the created mappings, we further identify those ones which are capable of producing unsatisfiable classes and therefore need to be debugged first. In the process of debugging, malicious bridge rules in mappings are automatically diagnosed and removed in accordance with the heuristic debugging discussed in Section 3. In the concluding step of the experimental study, we apply the minimization algorithm to compute for each mapping a logically-equivalent minimal set of bridge rules. Note that for those mappings which demand the debugging first the minimization is applied to their repaired descendants.

4.2 Results

The results of applying the heuristic debugging and minimization techniques to the automatically generated mappings are summarized in Table 1 and Table 2. More information about the test data and results can be obtained visiting the applications section of the DRAGO reasoner web page.[5]

During the debugging process we performed the following measurements: the initial amount of bridge rules in the mapping to be debugged, number of classes which become unsatisfiable due to the mapping, and finally the sets of bridge rules which are diagnosed as malicious and are automatically removed by the debugging algorithm. After the removal of malicious bridge rules, a mapping becomes repaired in a sense that it is not capable of producing unsatisfiability anymore. As shown in Table 1, the results of

---

[4] `http://nb.vse.cz/~svabo/oaei2006/`

[5] `http://sra.itc.it/projects/drago/applications.html`

Mapping	Bridge rules count	Unsatisfiable classes count	Removed bridge rules count	Set of removed bridge rules
CMT-CONFTOOL	48	3	3	$CMT : Conference \xrightarrow{\sqsubseteq} CONFTOOL : Organization$ $CMT : Person \xrightarrow{\sqsupseteq} CONFTOOL : Poster$ $CMT : ProgramCommitteeChair \xrightarrow{\sqsubseteq} CONFTOOL : Event$
CMT-CRS	53	1	1	$CMT : Document \xrightarrow{\sqsupseteq} CRS : program$
CMT-EKAW	116	4	5	$CMT : Person \xrightarrow{\sqsupseteq} Flyer$ $CMT : Person \xrightarrow{\sqsupseteq} Multi - author\_Volume$ $CMT : Person \xrightarrow{\sqsupseteq} Proceedings$ $CMT : ProgramCommitteeChair \xrightarrow{\sqsubseteq} Social\_Event$ $CMT : Person \xrightarrow{\sqsupseteq} Conference\_Proceedings$
CONFTOOL-CRS	80	10	15	$CONFTOOL : University \xrightarrow{\sqsubseteq} CRS : event$ $CONFTOOL : Social\_event \xrightarrow{\sqsupseteq} CRS : program$ $CONFTOOL : Author \xrightarrow{\sqsubseteq} CRS : event$ $CONFTOOL : Person \xrightarrow{\sqsupseteq} CRS : event$ $CONFTOOL : Participant \xrightarrow{\sqsubseteq} CRS : event$ $CONFTOOL : Event \xrightarrow{\sqsupseteq} CRS : participant$ $\ldots$
CRS-CMT	53	2	3	$CRS : document \xrightarrow{\sqsupseteq} CMT : Acceptance$ $CRS : document \xrightarrow{\sqsupseteq} CMT : ProgramCommitteeChair$ $CRS : program \xrightarrow{\sqsupseteq} CMT : ProgramCommitteeChair$
CRS-CONFTOOL	80	27	30	$CRS : conference \xrightarrow{\sqsubseteq} CONFTOOL : Organization$ $CRS : person \xrightarrow{\sqsubseteq} CONFTOOL : Event$ $CRS : person \xrightarrow{\sqsupseteq} CONFTOOL : Poster$ $CRS : document \xrightarrow{\sqsubseteq} CONFTOOL : Event$ $CRS : author \xrightarrow{\sqsubseteq} CONFTOOL : Event$ $CRS : participant \xrightarrow{\sqsubseteq} CONFTOOL : Event$ $CRS : event \xrightarrow{\sqsupseteq} CONFTOOL : Person$ $\ldots$
PCS-CONFTOOL	45	5	5	$PCS : Conference \xrightarrow{\sqsubseteq} CONFTOOL : Organization$ $PCS : Report \xrightarrow{\sqsubseteq} CONFTOOL : Event$ $PCS : Report \xrightarrow{\sqsubseteq} CONFTOOL : Organization$ $PCS : PERSON \xrightarrow{\sqsupseteq} CONFTOOL : Poster$ $PCS : Accepted\_paper \xrightarrow{\sqsubseteq} CONFTOOL : Event$
PCS-EKAW	120	4	5	$PCS : PERSON \xrightarrow{\sqsupseteq} EKAW : Flyer$ $PCS : PERSON \xrightarrow{\sqsupseteq} EKAW : Multi - author\_Volume$ $PCS : PERSON \xrightarrow{\sqsupseteq} EKAW : Proceedings$ $PCS : Web\_site \xrightarrow{\sqsubseteq} EKAW : Event$ $PCS : PERSON \xrightarrow{\sqsupseteq} EKAW : Conference\_Proceedings$
SIGKDD-CMT	60	1	1	$SIGKDD : Program\_Committee \xrightarrow{\sqsupseteq} CMT : ProgramCommitteeChair$
SIGKDD-CONFTOOL	72	3	3	$SIGKDD : Conference \xrightarrow{\sqsubseteq} CONFTOOL : Organization$ $SIGKDD : Person \xrightarrow{\sqsupseteq} CONFTOOL : Poster$ $SIGKDD : Deadline\_Author\_notification \xrightarrow{\sqsubseteq} CONFTOOL : Person$
SIGKDD-CRS	57	1	1	$SIGKDD : Document \xrightarrow{\sqsupseteq} CRS : program$
SIGKDD-EKAW	127	4	5	$SIGKDD : Person \xrightarrow{\sqsupseteq} EKAW : Flyer$ $SIGKDD : Person \xrightarrow{\sqsupseteq} EKAW : Multi - author\_Volume$ $SIGKDD : Person \xrightarrow{\sqsupseteq} EKAW : Proceedings$ $SIGKDD : Deadline\_Author\_notification \xrightarrow{\sqsubseteq} EKAW : Person$ $SIGKDD : Person \xrightarrow{\sqsupseteq} EKAW : Conference\_Proceedings$

Table 1. Debugging results.

Mapping	Bridge rules count	Entailed bridge rules count	Bridge rules reduction rate	Mapping	Bridge rules count	Entailed bridge rules count	Bridge rules reduction rate
CMT-CONFTOOL*	45	34	76%	EKAW-CMT	115	96	83%
CMT-CRS*	52	38	73%	EKAW-SIGKDD	127	95	75%
CMT-SIGKDD	59	45	76%	PCS-CONFTOOL*	40	25	63%
CMT-EKAW*	111	94	85%	PCS-CRS	38	21	55%
CONFTOOL-CMT	48	34	71%	PCS-SIGKDD	56	36	64%
CONFTOOL-CRS*	65	40	62%	PCS-CMT	73	58	79%
CONFTOOL-SIGKDD	75	43	57%	PCS-EKAW*	115	96	83%
CONFTOOL-PCS	45	27	60%	SIGKDD-CMT*	59	45	76%
CRS-CMT*	50	34	68%	SIGKDD-CONFTOOL*	69	41	59%
CRS-CONFTOOL*	50	37	74%	SIGKDD-CRS*	56	34	61%
CRS-SIGKDD	57	34	60%	SIGKDD-PCS	56	36	64%
CRS-PCS	38	21	55%	SIGKDD-EKAW*	122	94	77%

Table 2. Minimization results (starred mappings were first repaired applying the debugging).

applying the heuristic debugging approach proposed in Section 3 are quite reassuring – all of the mappings automatically removed by our method are actually incorrect ones.

To estimate minimization rate we measured the initial number of bridge rules and the amount of logically entailed bridge rules discovered by applying the minimization technique. As summarized in Table 2, the amount of the entailed bridge rules in a certain automatically generated mapping varies from 50 to 80$\%$ to the initial number of bridge rules in this mapping.

5 Discussion

We have presented a method for automatically improving the result of heuristic matching systems using logical reasoning. The basic idea is similar to existing work on debugging ontologies and uses some non-standard inference methods for reasoning about mappings introduced in previous work. The method feeds on the fact that most existing matching algorithms ignore the logical implications of new mappings. This gap is filled by our method that detects malicious impacts of generated mappings and traces them back to their source. As we have shown in the experiments, in almost all cases (in fact in all cases observed in the experiment) the unwanted effects were caused by wrong mappings and we were able to remove them automatically thus improving the correctness of the generated mapping. Actually, the idea of using logical reasoning in the matching process is not new and has been proposed by others (e.g., [7, 8]), the way it is used in our work, however, is unique, as it is the only approach that takes the effects of mappings into account. We believe that this additional step can significantly improve the quality of matching methods and should be integrated in existing matching algorithms as far as they are concerned with expressive ontologies that support consistency checking. In fact, the expressiveness of the language used to encode the ontologies to be matched seems to be the only limitation of our approach which can only be applied if the language supports consistency checking. In our experiments, we have seen that we can improve the correctness of matching results by removing wrong mappings. So far, we did not quantify this improvement, this has to be done in future work.

Acknowledgements

This work was partially supported by the German Science Foundation in the Emmy-Noether Program and by the European Union under grant FP6-507482 (KnowledgeWeb) as part of the T-Rex exchange program.

References

1. A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, 1:153–184, 2003.
2. P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou, and S. Tessaris. Specification of a common framework for characterizing alignment. Deliver. 2.2.4, KnowledgeWeb, 2004.
3. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. C-OWL: Contextualizing ontologies. In *Proceedings of the 2nd International Semantic Web Conference (ISWC-03)*, volume 2870 of *LNCS*, pages 164–179. Springer, 2003.
4. P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. In *Proceedings of the Second Internatinal Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 130–145. Springer Verlag, 2003.
5. D. Calvanese, G. De Giacomo, and M. Lenzerini. A framework for ontology integration. In *Proceedings of the Semantic Web Working Symposium*, pages 303–316, Stanford, CA, 2001.
6. M. Ehrig and S. Staab. QOM - Quick Ontology Mapping. In *Proceedings of the 3rd International Semantic Web Conference (ISWC-04)*, 2004.
7. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In *Proceedings of the European Semantic Web Conference (ESWS-04)*, pages 61–75, 2004.
8. F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient semantic matching. In *Proceedings of the European Semantic Web Conference (ESWS-05)*, pages 272–289, 2005.
9. E. Hovy. Combining and standardizing largescale, practical ontologies for machine translation and other uses. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC)*, pages 535–542, 1998.
10. J. Madhavan, P. A. Bernstein, P. Domingos, and A. Halevy. Representing and reasoning about mappings between domain models. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-02)*, 2002.
11. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering (ICDE-02)*. IEEE Computing Society, 2002.
12. N. F. Noy and M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
13. L. Serafini, A. Borgida, and A. Tamilin. Aspects of distributed and modular ontology reasoning. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, 2005.
14. L. Serafini, H. Stuckenschmidt, and H. Wache. A formal investigation of mapping languages for terminological knowledge. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, 2005.
15. L. Serafini and A. Tamilin. DRAGO: Distributed reasoning architecture for the semantic web. In *Proceedings of the 2nd European Semantic Web Conference (ESWC-05)*, 2005.
16. H. Stuckenschmidt, H. Wache, and L. Serafini. Reasoning about ontology mappings. In *Proceedings of the ECAI-06 Workshop on Contextual Representation and Reasoning*, 2006.
17. O. Svab, V. Svatek, P. Berka, D. Rak, and P. Tomasek. Ontofarm: Towards an experimental collection of parallel ontologies. In *Poster Proceedings of the International Semantic Web Conference 2005 (ISWC-05)*, 2005.

Results of the
Ontology Alignment Evaluation Initiative 2006 ⋆

Jérôme Euzenat[1], Malgorzata Mochol[2], Pavel Shvaiko[3], Heiner Stuckenschmidt[4],
Ondřej Šváb[5], Vojtěch Svátek[5], Willem Robert van Hage[6], and Mikalai Yatskevich[3]

[1] INRIA Rhône-Alpes, Montbonnot, France
`jerome.euzenat@inrialpes.fr`
[2] Free University of Berlin, Berlin, Germany
`mochol@inf.fu-berlin.de`
[3] University of Trento, Povo, Trento, Italy
`{pavel,yatskevi}@dit.unitn.it`
[4] University of Mannheim, Mannheim, Germany
`heiner@informatik.uni-mannheim.de`
[5] University of Economics, Prague, Czech Republic
`{svabo,svatek}@vse.cz`
[6] Vrije Universiteit Amsterdam, The Netherlands
`wrvhage@few.vu.nl`

Abstract. We present the Ontology Alignment Evaluation Initiative 2006 campaign as well as its results. The OAEI campaign aims at comparing ontology matching systems on precisely defined test sets. OAEI-2006 built over previous campaigns by having 6 tracks followed by 10 participants. It shows clear improvements over previous results. The final and official results of the campaign are those published on the OAEI web site.

1 Introduction

The Ontology Alignment Evaluation Initiative[7] (OAEI) is a coordinated international initiative that organizes the evaluation of the increasing number of ontology matching systems. The main goal of the Ontology Alignment Evaluation Initiative is to be able to compare systems and algorithms on the same basis and to allow anyone for drawing conclusions about the best matching strategies. Our ambition is that from such evaluations, tool developers can learn and improve their systems. The OAEI campaign is the evaluation of matching systems on consensus test cases.

Two first events have been organized in 2004: (i) the Information Interpretation and Integration Conference (I3CON) held at the NIST Performance Metrics for Intelligent Systems (PerMIS) workshop and (ii) the Ontology Alignment Contest held at the Evaluation of Ontology-based Tools (EON) workshop of the annual International

---

[⋆] This paper improves on the "First results" initially published in the on-site Ontology matching workshop proceedings. The only official results of the campaign, however, are on the OAEI web site.

[7] `http://oaei.ontologymatching.org`

Semantic Web Conference (ISWC) [6]. The first unique OAEI evaluation campaign has been presented at the workshop on Integrating Ontologies held in conjunction with the International Conference on Knowledge Capture (K-Cap) 2005 [1]. The campaign of 2006 is presented at the Ontology Matching (OM) workshop at ISWC, in Athens, Georgia, USA.

In reaction over last year's remarks, this year we have a variety of test cases that emphasize different aspects of the matching needs. From three test cases last year, we now have six very different test cases. Some of these tests introduce particular modalities of evaluation, such as a consensus building workshop and application-oriented evaluation.

This paper serves as an introduction to the evaluation campaign of 2006 and to the results provided in the following papers. The remainder of the paper is organized as follows. In Section 2 we present the overall testing methodology that has been used. Sections 3-8 discuss in turn the settings and the results of each of the test cases. Section 9 overviews lessons learned based on the campaign. Finally, Section 10 outlines future plans and Section 11 concludes.

2 General methodology

We present the general methodology for the 2006 campaign as it was defined and report its execution.

2.1 Test cases

This year's campaign has consisted of four tracks gathering six data sets and different evaluation modalities.

Comparison track: benchmark (§3) Like in previous campaigns, a systematic benchmark series has been produced. The goal of this benchmark series is to identify the areas in which each matching algorithm is strong or weak. The test is based on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided.

Expressive ontologies

- **anatomy (§4)** The anatomy real world case covers the domain of body anatomy and consists of two ontologies with an approximate size of several 10k classes and several dozens of relations.
- **jobs (§5)** The jobs test case is an industry evaluated real world business case. A company has a need to improve job portal functionality with semantic technologies. To enable higher precision in retrieval of relevant job offers or applicant profiles, OWL ontologies from the employment sector are used to describe jobs and job seekers and matching with regard to these ontologies provides the improved results. For confidentiality reasons, the test is run by the company team with software provided by the participants.

Directories and thesauri

- **directory (§6)** The directory real world case consists of matching web directories, such as open directory, Google and Yahoo. It has more than 4 thousands of elementary tests.
- **food (§7)** Two SKOS thesauri about food have to be aligned using relations from the SKOS Mapping vocabulary. All results are evaluated by domain experts. Each participant is asked to evaluate a small part of the results of the other participants.

Consensus workshop: conference (§8) Participants have been asked to freely explore a collection of conference organization ontologies (the domain being well understandable for every researcher). This effort was expected to materialize in usual alignments as well as in interesting individual correspondences ("nuggets"), aggregated statistical observations and/or implicit design patterns. There is no a priori reference alignment. For a selected sample of correspondences, consensus was sought at the workshop and the process of reaching consensus was recorded.

Table 1 summarizes the variation in the results expected from these tests.

test	language	relations	confidence	modalities
benchmarks	OWL	=	[0 1]	open
anatomy	OWL	=	1	blind
jobs	OWL	=	[0 1]	external
directory	OWL	=	1	blind
food	SKOS	narrowMatch, exactMatch, broadMatch	1	blind+consensual
conference	OWL-DL	$=, \leq$	1	blind+consensual

Table 1. Characteristics of test cases (open evaluation is done with already published expected results, blind evaluation is done by organizers from reference alignments unknown to the participants, consensual evaluation is obtained by reaching consensus over the found results and external evaluation is preformed independently of the organizers by running the actual systems).

2.2 Preparatory phase

The ontologies and alignments of the evaluation have been provided in advance during the period between June 1st and June 28th. This gave potential participants the occasion to send observations, bug corrections, remarks and other test cases to the organizers. The goal of this preparatory period is to be sure that the delivered tests make sense to the participants. The tests still evolved after this period, but only for ensuring a better participation to the tests. The final test base has been be released on August 23rd.

2.3 Execution phase

During the execution phase the participants used their systems to automatically match the ontologies from the test cases. Participants have been asked to use one algorithm and the same set of parameters for all tests in all tracks. It is fair to select the set of parameters that provide the best results (for the tests where results are known). Beside parameters, the input of the algorithms must be the two ontologies to be aligned and any general purpose resource available to everyone, i.e., no resource especially designed for the test. In particular, the participants should not use the data (ontologies and reference alignments) from other test sets to help their algorithms.

Ontologies are, in most cases, described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in the Alignment format expressed in RDF/XML. All the participants also provided the papers that are published hereafter and a link to their program and its configuration parameters.

2.4 Evaluation phase

The organizers have evaluated the results of the algorithms used by the participants and provided comparisons on the basis of the provided alignments.

In order to ensure that it is possible to process automatically the provided results, the participants have been requested to provide (preliminary) results by September 4th. In the case of blind tests only the organizers did the evaluation with regard to the withheld reference alignments. In the case of double blind tests, the participants provide a version of their system and the values of the parameters if any.

The standard evaluation measures are precision and recall computed against the reference alignments. For the matter of aggregation of the measures we use weighted harmonic means (weights being the size of the true positives). This clearly helps in case of empty alignments. Another technique that has been used is the computation of precision/recall graphs so it was advised that participants provide their results with a weight to each correspondence they found.

New measures addressing some limitations of precision and recall have also been used for testing purposes. These were presented at the workshop discussion in order for the participants to provide feedback on the opportunity to use them in a further evaluation.

2.5 Comments on the execution

This year again, we had more participants than in previous years: 4 in 2004, 7 in 2005 and 10 in 2006. We also noted the increase in tools compliance and robustness: they had less problems to carry the tests and we had less problems to evaluate the results.

We have had not enough time so far to validate the results which have been provided by the participants. Last year, validating these results has proved feasible so we plan to do it again in the future (at least for those participants who provided their systems).

We summarize the list of participants in Table 2. Similar to last year not all participants provided results for all tests. They usually did those which are easier to run, such as benchmark, directory and conference. The jobs line corresponds to the participants who have provided an executable version of their systems. The variety of tests and the short time given to provide results have certainly prevented participants from considering more tests.

test \ system	falcon	hmatch	dssim	coma	automs	jhuapl	prior	RiMOM	OCM	nih	Total
benchmark	√	√	√	√	√	√	√	√	√		9
anatomy	√	√		√			√			√	5
jobs	√	√		√	√		√		√		6
directory	√	√		√	√		√	√	√		7
food	√	√		√			√	√			5
conference	√	√		√	√			√	√		6
certified confidence	√	√		√			√	√			5
time					√		√	√	√	√	5

Table 2. Participants and the state of their submissions. Confidence is ticked when given as non boolean value. Time indicates when participants included execution time with their tests.

Like last year, the time devoted for performing these tests (three months) and the period allocated for that (summer) is relatively short and does not really allow the participants to analyze their results and improve their algorithms. On the one hand, this prevents having algorithms to be particularly tuned for the tests. On the other hand, this can be frustrating for the participants. The timeline is very difficult to handle, hence, we should try to give more time for the next campaign.

The summary of the results track by track is provided in the following six sections.

3 Benchmark

The goal of the benchmark tests is to provide a stable and detailed picture of each algorithm. For that purpose, the algorithms are run on systematically generated test cases.

3.1 Test set

The domain of this first test is Bibliographic references. It is, of course, based on a subjective view of what must be a bibliographic ontology. There can be many different classifications of publications, for example, based on area and quality. The one chosen here is common among scholars and is based on publication categories; as many ontologies (tests #301-304), it is reminiscent to BibTeX.

The systematic benchmark test set is built around one reference ontology and many variations of it. The reference ontology is that of test #101.The participants have to match this reference ontology with the variations. These variations are focusing the characterization of the behavior of the tools rather than having them compete on real-life problems. The ontologies are described in OWL-DL and serialized in the RDF/XML format. This reference ontology contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals.

Since the goal of these tests is to offer some kind of permanent benchmarks to be used by many, the test is an extension of the 2004 EON Ontology Alignment Contest.

The reference ontology has been improved last year by including circular relations that were missing from the first test. In 2006, we have put the UTF-8 version of the tests as standard, the ISO-8859-1 being optional. Test numbering (almost) fully preserves the numbering of the first EON contest.

The kind of expected alignments is still limited: they only match named classes and properties, they mostly use the "=" relation with confidence of 1.

There are still three groups of tests in this benchmark:

- simple tests (1xx) such as comparing the reference ontology with itself, with another irrelevant ontology (the wine ontology used in the OWL primer) or the same ontology in its restriction to OWL-Lite;
- systematic tests (2xx) that were obtained by discarding some features from some reference ontology. It aims at evaluating how an algorithm behaves when this information is lacking. The considered features were:
  Name of entities that can be replaced by random strings, synonyms, name with different conventions, strings in another language than english,
  Comments that can be suppressed or translated in another language,
  Specialization Hierarchy ăthat can be suppressed, expanded or flattened,
  Instances that can be suppressed,
  Properties that can be suppressed or having the restrictions on classes discarded, and
  Classes that can be expanded, i.e., replaced by several classes or flattened.
- four real-life ontologies of bibliographic references (3xx) that were found on the web and left mostly untouched (there were added xmlns and xml:base attributes).

Full description of these tests can be found on the OAEI web site.

3.2 Results

Table 3 provides the consolidated results, by groups of tests. We display the results of participants as well as those given by some very simple edit distance algorithm on labels (edna). Like last year, the computed values are real precision and recall and not a simple average of precision and recall. The full results are on the OAEI web site.

These results show already that three systems are relatively close (coma, falcon and RiMOM). The RiMOM system is slightly ahead of the others on these raw results. The DSSim system obviously favoured precision over recall but its precision degrades with “real world” 3xx series. No system had strictly lower performance than edna.

The results have also been compared with the three measures proposed in [3] last year (symmetric, effort-based and oriented). These are generalisation of precision and recall in order to better discriminate systems that slightly miss the target from those which are grossly wrong. The three measures provide the same results. This is not really surprising given the proximity of these measures. As expected, they can only improve over traditional precision and recall. The improvement affects all the algorithms, but this is not always strong enough for being reflected in the aggregated results. Moreover, the new measures do not dramatically change the evaluation of the participating systems.

Each algorithm has its best score with the 1xx test series. There is no particular order between the two other series. Again, it is more interesting to look at the 2xx

algo	refalign		edna		automs		coma		DSSim		falcon		hmatch		jhuapl		OCM		prior		RiMOM	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	1.00	1.00	0.96	1.00	0.94	1.00	1.00	1.00	1.00	0.98	1.00	1.00	0.91	1.00	1.00	1.00	0.95	1.00	1.00	1.00	1.00	1.00
2xx	1.00	1.00	0.90	0.49	0.94	0.64	0.96	0.82	0.99	0.49	0.91	0.85	0.83	0.51	0.20	0.86	0.93	0.51	0.95	0.58	0.97	0.87
3xx	1.00	1.00	0.94	0.61	0.91	0.70	0.84	0.69	0.90	0.78	0.89	0.78	0.78	0.57	0.18	0.50	0.89	0.51	0.85	0.80	0.83	0.82
Total	1.00	1.00	0.91	0.54	0.94	0.67	0.96	0.83	0.98	0.55	0.92	0.86	0.84	0.55	0.22	0.85	0.93	0.55	0.95	0.63	0.96	0.88
Symmetric	1.00	1.00	0.91	0.54	0.94	0.68	0.96	0.83	0.99	0.55	0.94	0.89	0.85	0.56	0.22	0.87	0.93	0.55	0.96	0.64	0.97	0.89
Effort-based	1.00	1.00	0.91	0.54	0.94	0.68	0.96	0.83	0.99	0.55	0.94	0.89	0.85	0.56	0.22	0.87	0.93	0.55	0.96	0.64	0.97	0.89
Oriented	1.00	1.00	0.91	0.54	0.94	0.68	0.96	0.83	0.99	0.55	0.94	0.89	0.85	0.56	0.22	0.87	0.93	0.55	0.96	0.64	0.97	0.89

Table 3. Means of results obtained by participants on the benchmark test case (corresponding to harmonic means).

series structure to distinguish the strengths of algorithms. This will be done in a separate document.

This year the apparently best algorithms provided their results with confidence measures. It is thus possible to draw precision/recall curves in order to compare them. We provide in Figure 1 the precision and recall graphs of this year. They involve only the results of participants who provided confidence measures different of 1 or 0 (see Table 2). They also feature the results for edit distances on class names (edna) and the results of Falcon last year (falcon-2005). The graph for falcon2005 is not really accurate since falcon2005 provided 1/0 alignments last year. This graph has been drawn with only technical adaptation of the technique used in TREC. Moreover, due to lack of time, these graphs have been computed by averaging the graphs of each of the tests (instead to pure precision and recall).

Contrary to last year, we have three systems competing at the higest level (falcon, coma and RiMOM) and a gap between these and the next systems. No system is significantly outperformed by standard edit distance (edna). The best systems are at the level of last year's best system (falcon).

Like last year we have compared the results of this year's systems with the previous years on the basis of the 2004 tests. (Table 4). The three best systems (falcon, coma and RiMOM) arrive at the level of last year's best system (falcon). However, no system outperforms it.

Unfortunately no representant of the group of systems that followed falcon last year is present this year.

Year	2004				2005		2006					
System	fujitsu		stanford		falcon		RiMOM		falcon		coma	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2xx	0.93	0.84	0.98	0.72	0.98	0.97	1.00	0.98	0.97	0.97	0.99	0.97
3xx	0.60	0.72	0.93	0.74	0.93	0.83	0.83	0.82	0.89	0.78	0.84	0.69
H-means	0.88	0.85	0.98	0.77	0.97	0.96	0.97	0.96	0.97	0.95	0.98	0.94

Table 4. Evolution of the best scores over the years (on the basis of 2004 tests).

4 Anatomy

The focus of the anatomy test case is to confront existing matching technology with real world ontologies. Our aim is to get a better impression of where we stand with respect to really hard challenges that normally require an enormous manual effort and in-depth knowledge of the domain.

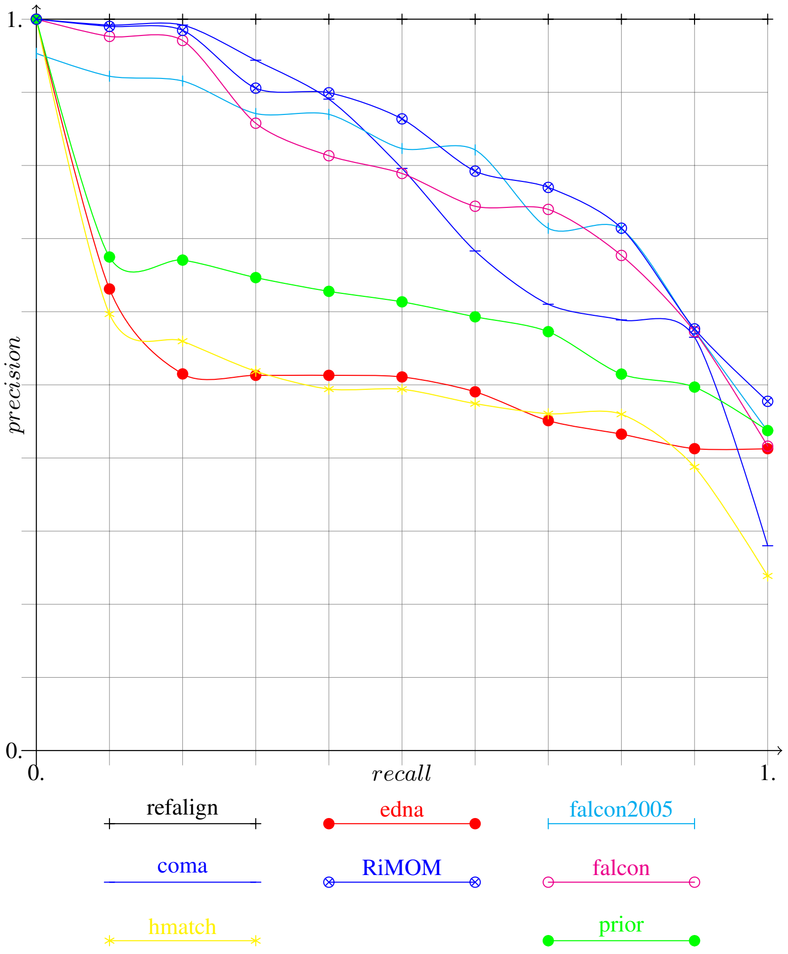


Fig. 1. Precision/recall graphs for the systems which provided confidence values in their results.

4.1 Test set

The task is placed in the medical domain as this is the domain where we find large, carefully designed ontologies. The specific characteristics of the ontologies are:

- Very large models: OWL models of more than 50MB.
- Extensive class hierarchies: Ten thousands of classes organized according to different views on the domain.
- Complex relationships: Classes are connected by a number of different relations.
- Stable terminology: The basic terminology is rather stable and should not differ too much in the different models.
- Clear modeling principles: The modeling principles are well-defined and documented in publications about the ontologies.

As a consequence, the anatomy test set actually tests existing matching systems with respect to two questions.

- Do existing approaches scale to very large models?
- Are existing approaches able to take advantage of well-documented modeling principles and knowledge about the domain?

This also means that the goal of this test case is not to compare the performance of matching systems on a quantitative basis. We are rather interested in how many systems are actually able to create mappings at all and in specific heuristics used for computing matches that are described in the corresponding papers.

The ontologies to be aligned are different representations of human anatomy developed independently by teams of medical experts. Both ontologies are available in OWL format and mostly contain classes and relations between them. The use of axioms is limited.

The Foundational Model of Anatomy The Foundational Model of Anatomy has been developed by the University of Washington. It is an ontology describing the human anatomy including a taxonomy of body parts, information about anatomical structures and structure transformations. According to the developers the Foundational Model of Anatomy ontology contains approximately 75.000 classes and over 120.000 terms; over 2.1 million relationship instances from 168 relationship types link the FMA's classes into a coherent symbolic model.

We extracted an OWL version of the ontology from a Protégé database. The resulting model is in OWL-full as relations are defined between classes rather than instances.

Galen The second ontology is the anatomy model developed in the OpenGalen Project by the University of Manchester. According to the creators, the ontology contains around 10.000 concepts covering a bit more than standard textbook anatomy in terms of body parts, anatomical structures and relations between different parts and structures.

The ontology is freely available as a Protégé Project file on the OpenGalen web page. We created an OWL version of the ontology using the export functionality of Protégé. The resulting ontology is in OWL-DL thus supporting logical reasoning about inconsistencies.

4.2 Results

The anatomy use case is part of the ontology alignment evaluation challenge for the second time now. While in 2005, none of the participants was in the position to submit a result for this data set. Almost all participants reported major difficulties in processing the ontologies due to their size and the fact that one of the models is in OWL full. At least these scalability problems seem to be solved this year. In the 2006 campaign, five out of ten participants submitted results for the anatomy data set. This clearly shows the advance of matching systems on the technical level and also shows that matching technologies are ready for large scale applications.

On the content level the results are much harder to judge. Due to the lack of a reference mapping, we were not able to provide a quantitative judgement and comparison of the different systems. In the evaluation, we rather concentrated on the coverage of the ontologies, the degree of agreement amongst the matching systems and on the specific techniques used the matching systems to address this challenging alignment task.

A first observation, we made was that none of the systems managed to reach a good coverage of the ontologies. Although both models contain several ten thousand concepts and the fact that we can assume a high degree of overlap in the two models, the systems were only able to produce mappings for 2000 to 3000 concepts, which is less than 5% of the concepts in the FMA.

We also found out that systems have severe difficulties with irregular concept names. The GALEN ontology contains a subset of concepts with highly irregular concept names. It turned out that only one system (Coma++) was able to determine mappings for these concepts – for the price of not being able to match any of the concept names with regular names.

Looking at the actual methods used by the systems we see a common pattern. Almost all systems use the linguistic similarity between class names and other features of the class description as a basis for determining candidates. Normally, the systems combine different similarity measures. On top of this purely linguistic comparison, some systems also apply structural techniques. In particular, they translate the models into a graph structure and propagate the individual similarity in the graph structure. Only one of the systems (NIH) actually used reasoning techniques to validate hypothesis and to determine matches based on the semantics of the models.

4.3 Discussion and Conclusions

For the first time since the anatomy data set has been used in the ontology alignment evaluation challenge, we are in a position, were we were actually able to compare the results of different matching systems. The results show that there is still a lot of work to do to make matching systems ready for real life applications. The problems above showed that differences in the naming scheme of classes can already cause matchers to fail on a significant subset of the vocabulary. It seems that existing matchers suffer from the need to balance precision and recall in determining mappings. This conclusion is backed by results from other experiments, where it turned out that matching systems that produce highly precise mappings miss many mappings found by other systems. We conclude that using fixed thresholds to determine mapping candidates is not a good way for trading-off precision and recall.

We were disappointed to see that only one system actually used some form of reasoning in order to take the meaning of the ontologies into account. As one of the major advantages of OWL is the ability to specify and reason about the semantics of concepts, it is at least surprising that this feature is not exploited by existing matchers. In fact, logical reasoning could be a way of becoming less dependent on the quality of certain similarity measures that obviously have some limitations when it comes to complex ontologies.

In summary, the results of the anatomy test case have shown that there is some significant progress in terms of the maturity of matching technology. On the other hand, the results also show that there are still a lot of open problems with respect to producing good alignments on real life cases. For the setup of the next challenge this means that we have to think about a more precise evaluation of the matching results in order to determine where exactly the problems of different matchers are. For this purpose it is necessary to have a reference alignment to compare against. Currently there are two possible ways to make such an evaluation possible. The first option is to move to a different, but related data set for which a reference mapping exists. Such a data set exists in terms of the anatomy part of the NCI thesaurus and the Adult Mouse Anatomy ontology. These ontologies would be much smaller in size but support a quantitative evaluation. The other option is to start building a reference mapping for the current data set using the mappings created by the participants of this years challenge as a starting point.

5 Jobs

The goal of the job test case is to evaluate the results of matching ontologies in the application context.

5.1 Test set

Semantic web technologies are used to semantically annotate job postings and applicant profiles in order to increase market transparency together with avoiding the bottleneck of a central database. In a semantic recruitment application the data exchange between employers, job applicants and job portals is based on a set of shared vocabularies describing domain relevant terms: occupations, industrial sectors and skills. These commonly used vocabularies have been formally defined by means of a Human Resource ontology (HR-ontology). The implementation of the HR-ontology was realized by translating several semi-structured input formalisms and encoding text-based classification standards into OWL. This ontology is used in a job matching application for computing the similarity between jobs and profiles. The current application uses an algorithm which is based on the similarity between two concepts determined by the distance between them.

We planned to modify this application in order that it can take advantage of the alignments found by participants to compute similarity. The matching systems as well as their parameters have been provided to the organizers who could run the algorithms on the ontologies and obtain the alignments. These alignments were to be used by job matchers in order to compare 250 job offers with about 250 applicant profiles.

5.2 Results

The results are not available at the time of writing. They will be made available on the OAEI web site if we can find time to complete this test.

6 Directory

The directory test case aims at providing a challenging task for ontology matchers in the domain of large directories.

6.1 Test set

The data set exploited in the web directories matching task was constructed from Google, Yahoo and Looksmart web directories as described in [2; 7]. The dataset is presented as taxonomies where the nodes of the web directories are modeled as classes and classification relation connecting the nodes was modeled as rdfs:subClassOf relation.

The key idea of the data set construction methodology is to significantly reduce the search space for human annotators. Instead of considering the full mapping task which is very big (Google and Yahoo directories have up to $3 * 10^5$ nodes each: this means that the human annotators need to consider up to $(3 * 10^5)^2 = 9 * 10^{10}$ mappings), it uses semi automatic pruning techniques in order to significantly reduce the search space. For example, for the dataset described in [2], human annotators consider only 2265 mappings instead of the full mapping problem.

So, there has been 3 proposed representation for this year:

- one matching task between two taxonomies of 10^3 categories (full test set),
- one matching task between two taxonomies of 10^2 categories (10% test set), and
- 4639 matching task between two paths of around 10 categories (unit test sets).

The first data set incorporates the matching tasks involved in the unit tests which also correspond to the reference set. The second data set is guarantee to contain 10% of these unit tests.

This year the reference data set has been significantly extended with respect to the one exploited in OAEI-2005 [4]. In particular, the reference mapping contains not only positive but also negative mappings, which are used to approximate not only recall but also precision. The key difference of the reference mapping with respect to conventional ones, such as ones exploited in benchmark tests in this evaluation, is that it does not contain the complete set of reference mappings (R). Instead of this the reference mapping is composed of two parts [7]:

- Representative subset of complete reference mapping ($P \subseteq R$). It contains the positive mappings, i.e., the mappings that hold for the matching task.
- Representative subset of negative mappings ($N \subseteq \bar{R}$), i.e., the mappings that do not hold for the matching task.

The reference mapping is composed of 2265 positive and 2374 negative mappings. Therefore the matching unit test set corresponds to 2265+2374=4639 tasks of finding the semantic relation holding between paths to root in the web directories modeled as sub class hierarchies.

6.2 Results

Approximate precision, recall and F-measure of the systems on web directories dataset are presented on Figure 2, 3 and 4 respectively. Given an alignment A and the set P and N of positive and negative mappings, approximate precision and recall are computed by:

$$AP(A, P, N) = \frac{|P \cap A|}{|P \cap A| + |N \cap A|} \qquad AR(A, P) = \frac{|P \cap A|}{|P|}$$

These formula, especially that of AP, generalize precision and recall by not taking the whole set of valid correspondences as reference alignment. They are called approximate precision and recall because when $P = R$ and $N = \bar{R}$ (R is the reference alignment), they correspond to precision and recall. How this is an accurate approximation of precision and recall heavily depends on the choice of P and N; these are discussed in [2; 7].

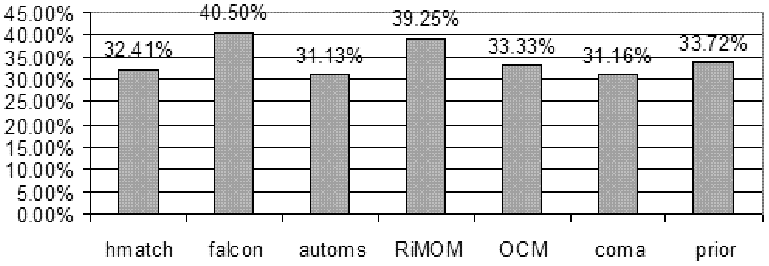


Fig. 2. Approximate precision for web directories matching task.

Similarly with OAEI-2005, 7 matching systems were evaluated on the dataset. However, only one of them (Falcon) participated in both evaluations. The systems in general demonstrated higher results than in OAEI-2005. The average approximate recall of the systems increased from 22.23% to 25.82%. The highest approximate recall (45.47%) was demonstrated by the Falcon system what is almost 50% increase in respect to its last year result (31.17%).

Despite of this progress the dataset remains difficult for the matching systems. The maximum and average values for approximate precision (40.5% and 34.5%), approximate recall (45.47% and 25.82%) and approximate F-measure (42.85% and 28,56%) are significantly lower than corresponding real values in benchmark tests for example.

Partition of positive and negative mappings according to the systems results are presented on Figure 5 and 6.

As from the figures, 43% of positive mappings have not been found by any of the systems. At the same time 22% of negative mappings were found by all the matching systems, i.e., all the matching systems mistakenly returned them as positive. Moreover only 10% of positive mappings were found by all the matching systems.

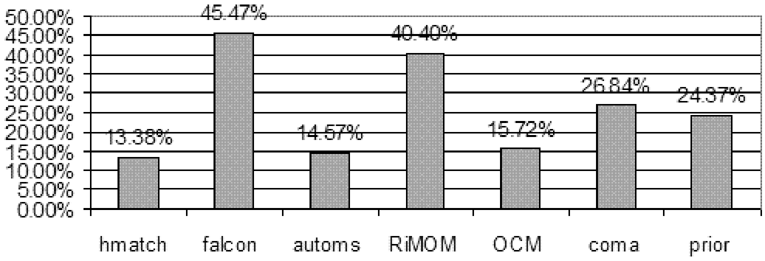


Fig. 3. Approximate recall for web directories matching task.

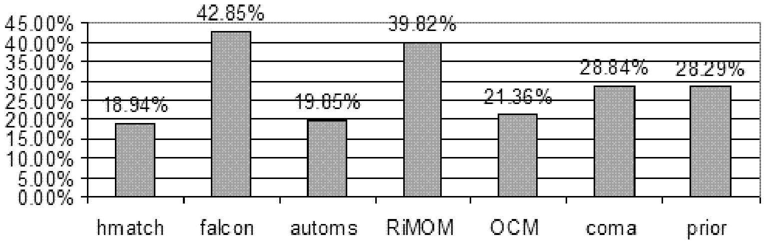


Fig. 4. Approximate F-measure for web directories matching task.

6.3 Comments

Six out of seven systems that participated in the evaluation presented their results only for one of the dataset representations, namely for the representation composed from 4639 node matching tasks. Only one system (H-Match) presented the results also for the other representations. Since, the other tasks were proposed in order to test scalability of the approaches, this can be interpreted as a sign of poor scalability of the systems participating in the evaluation.

Blind evaluation declined for some of the systems a possibility to improve their final results after preliminary result disclosure. For example, the final results of the coma and prior matching systems were slightly lower than their preliminary results. The final F-measure of coma dropped from 32.56% to 28.84% while F-measure of prior dropped from 28.32% to 28.29%.

7 Food

The food test case is another taxonomy task in which the taxonomies are taken out of theauri, i.e., they have a lot of text involved compared to the previous test case, and they are expressed in SKOS.

7.1 Test set

The task of this case consists of matching two thesauri formulated in SKOS:

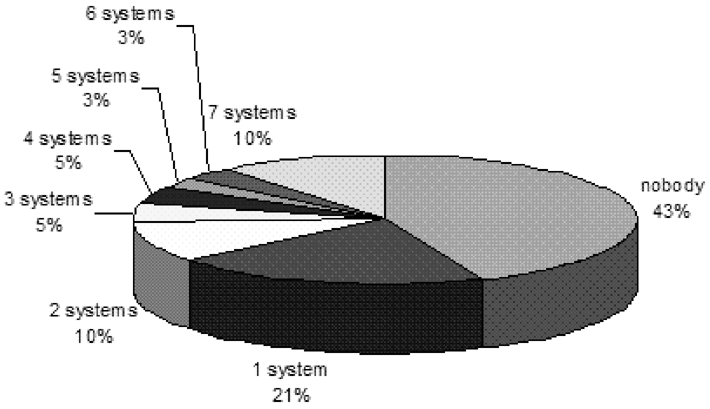


Fig. 5. Partition of the systems results on positive mappings.

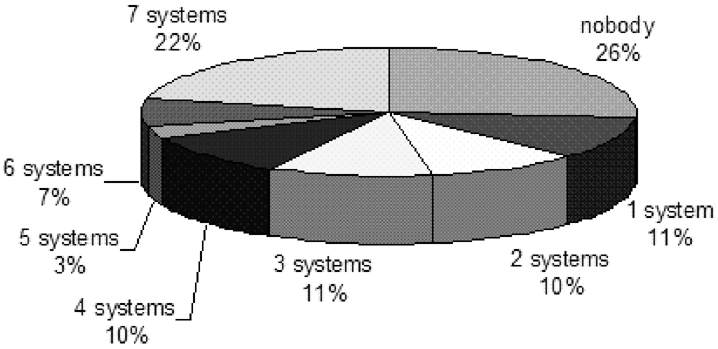


Fig. 6. Partition of the systems results on negative mappings.

AGROVOC The United Nations Food and Agriculture Organization (FAO) AGROVOC thesaurus, version May 2006. This thesaurus consists of 28.174 descriptor terms, i.e., prefered terms, and 10.028 non-descriptor terms, i.e., alternative terms. AGROVOC is multilingual in ten languages (en, fr, es, ar, zh, pt, cs, ja, th, sk).

NALT The United States National Agricultural Library (NAL) Agricultural thesaurus, version 2006. This thesaurus consists of 41.577 descriptor terms and 24.525 non-descriptor terms. NALT is monolingual, English.

Participants had to match these SKOS versions of AGROVOC and NAL using the exactMatch, narrowMatch, and broadMatch relations from the SKOS Mapping Vocabulary.

7.2 Evaluation procedure

Five participants took part in the OAEI 2006 food alignment task, South East University (Falcon-AO), University of Pittsburgh (Prior), Tsinghua University (RiMOM), University of Leipzig (COMA++), and Universitá degli Studi di Milano (HMatch). Each team provided between 10.000 and 20.000 alignments. This amounted to 31.112 unique alignments in total.

In order to give dependable precision results within the time span of the alignment initiative given a limited number of assessors we did a sample evaluation on 7% of the alignments. This sample was chosen to be representative of the type of topics covered by the thesauri and to be impartial to each participant and impartial to how much consensus amongst the participants there was about each alignment. We distinguished three categories of topics in the thesauri that each required a different level of domain knowledge of the assessors: Taxonomical concepts (plants, animals, bacteria, etc.), biological and chemical terms (structure formulas, terms from generics, etc.), and the remaining concepts (geography, agricultural processes, etc.). Under the authority of taxonomists at the US Department of Agriculture the taxonomical category of mappings was assessed using the strict rules that apply to the naming scheme of taxonomy. These are that if the preferred term of one concept is exactly the same as either the preferred or the alternative term of another concept then the concepts are considered to be exact matches. The latter two categories were assessed by two groups, a group of domain experts from the USDA and the FAO, and a group of computer scientists at the EKAW conference. The agreement between these groups was 72%. The computer scientists were less likely to judge an alignment to be correct than the domain experts.

As a significance test on precision scores of the systems we used the Bernoulli distribution. The precision of system A, P_A can be considered to be significantly greater than that of system B for a sample set of size N (in the cases of the three categories we distinguished respectively 18.399, 250, and 650) when the following formula holds:

$$|P_A - P_B| > 2\sqrt{\left(\frac{P_A \cdot (1 - P_A)}{N}\right)^2 + \left(\frac{P_B \cdot (1 - P_B)}{N}\right)^2}$$

Giving dependable recall numbers within the time span of the alignment initiative was not feasible, so we estimated recall on four sample sub-hierarchies of the thesauri: All oak trees (everything under the concept representing the Quercus genus), All rodents (everything under Rodentia), Geographical concepts of Europe, and everything under the NALT concept animal health and all AGROVOC concepts that have alignments to these concepts and their sub-concepts. These four samples respectively have size 41, 42, 74, and 34. Around 30% of the mappings were broadMatch and narrowMatch, the rest was exactMatch.

7.3 Results

The taxonomical parts of the thesauri accounted for by far the largest part of the alignments. The more difficult alignments that required lexical normalization, such as structure formulas, and relations that required background knowledge, such as many of the relations in the miscellaneous domain, accounted for a smaller part of the alignment. This caused systems that did well at the taxonomical mappings to have a great advantage over the other systems. The Falcon-AO system performed consistently best at the largest of the two categories and thus achieved high precision.

All systems only returned `exactMatch` alignments. This means that recall of all systems was limited to 71%. The RiMOM system managed to discover more good results than the Falcon-AO system on the four small sample recall bases, at the cost of some precision. Since recall was assessed on such a small set of examples we can only draw conclusions based on the precision results, but if the difference in recall between RiMOM and Falcon-AO persists throughout the rest of them, RiMOM achieves a better F-measure than Falcon-AO.

	RiMOM	Falcon-AO	Prior	COMA++	HMatch
Precision (taxonomical)	82%	83%$\star$	68%	43%	48%
Precision (bio/chem)	85%$\star$	80%	81%	76%	83%
Precision (miscellaneous)	78%	83%$\star$	74%	70%	80%
Precision (all topics)	**81%**	**83%**$\star$	**71%**	**54%**	**61%**

Table 5. Precision results based on sample evaluation. $\star$ indicates the significantly best system.

	RiMOM	Falcon-AO	Prior	COMA++	HMatch
Recall (all relations)	**50%**	**46%**	**45%**	**23%**	**46%**
Recall (only `exactMatch`)	71%	65%	64%	33%	65%

Table 6. Tentative estimation of recall based on sample evaluation.

	RiMOM	Falcon-AO	Prior	COMA++	HMatch
F-measure (all rel. & top.)	**62%**	**59%**	**55%**	**33%**	**53%**

Table 7. Tentative estimation of F-measure based on sample evaluation.

alignment found by # systems	1	2	3	4	5
average precision	6%	35%	67%	86%	99%
# alignments	21.663	2.592	2.470	4.467	5.555

Table 8. Consensus: average precision of the alignments returned by a number of systems.

8 Conference

The conference test set introduces matching several ontologies together as well as a consensus workshop aiming at studying the elaboration of consensus when establishing reference alignments.

8.1 Test set

The collection consists of ten ontologies in the domain of organizing conferences. The main features of this test set are:

- *Generally understandable domain.* Most ontology engineers are familiar with organizing conferences. Therefore, they can create their own ontologies as well as evaluate the mapping among their concepts with enough erudition.
- *Independence of ontologies.* Ontologies were developed independently and based on different resources, they thus capture the issues in organizing conferences from different points of view and with different terminology.
- *Relative richness in axioms.* Most ontologies were equipped with DL axioms of various kinds, which opens a way to use semantic matchers.

Ontologies differ in numbers of classes, of properties, in their DL expressivity, but also in underlying resources. Six ontologies are based *on tools* supporting the task of organizing conferences, two are based on experience of people with *personal participation* in conference organization, and two are based on *web pages* of concrete conferences.

8.2 Results

For the sake of brevity, all results from the initial evaluation phase are on the result report page[8]. There you can find global statistics about participants' results, which more or less reflect their quality. Additional, finer-grained results were obtained at the "consensus building workshop".

As there was no reference alignment to compare with, only a general statistics of submissions plus some simple observations were available to the date of writing this material. The statistics (counts of ontology pairs processed) follow:

- Automs tried to map all ontologies to three ontologies (30 alignments).
- Coma and Falcon delivered 45 alignments, i.e., all ontologies were mapped to each other.
- In the case of Hmatch, 90 pairs of ontologies were mapped, separately including each direction of mapping.
- RiMOM mapped all 100 pairs of ontologies, separately including each direction of mapping. Mapping of ontology onto itself was also included.
- In the case of OCM, 21 pairs of ontologies were mapped. Some ontologies were omitted because of their high complexity.

---

[8] http://nb.vse.cz/ svabo/oaei2006/

Other comments:

- Only equivalence, i.e., no subsumption, relations were discovered; for concepts and for properties separately, not across.
- Four participants delivered correspondences with certainty factors between 0 and 1 (coma, falcon, hmatch and RiMOM); the two remaining ones (automs and OCM) delivered ‘certain’ correspondences.
- One associated OAEI paper, by the AUTOMS team, discussed the role of different techniques (string matching, structure matching, thesaurus term matching) for different correspondences.
- Independently from OAEI, the conference collection has been investigated with the method of [5]. They evaluated the alignments of four systems, only among concepts. On the base of their evaluation, Falcon outperforms others in terms of precision (based on single person judgment). Their evaluation was also discussed at the consensus workshop.

Consensus building workshop During the “Ontology matching” workshop we organized a “Consensus building workshop”. The main idea behind this workshop was to thoroughly discuss controversial mappings, i.e., those on which tools disagree, and thus partly provide *feedback* for authors of involved systems and partly examine the *argumentation process*. Altogether 9 mappings were discussed, chosen before the workshop by the organizers, and the group finally achieved consensus for each mapping. A presentation contains both the information about evaluation and discussed “controversial” mappings[9]. Chosen mappings as candidates of controversy were representatives of following phenomena:

- subsumption - Two elements are considered as equivalent by the systems, but they are rather in relation of subsumption, e.g., pairs: `Document` vs. `article` and `Location` vs. `Place`.
- inverse property - Pair of elements are considered as equivalent by systems, but they are inverse, e.g., `reviews` vs. `hasReview`.
- lexical confusion – This category contains such mappings considered by systems that are wrong and mainly based on lexical similarity, e.g., `PC_Member` vs. `Member_PC`.
- Other phenomena that were not included in the choice for discussion are for example siblings (elements are rather siblings) and heterogenous mappings (matching of relation to class or vice versa).

Regarding *arguments* against and for, we experience that *lexical* reasons of mapping are first considered. Then follow arguments with regard to *context* of elements in question. This means consideration of certain neighborhood, subclasses and superclasses (in the case of properties, we can consider subproperties and superproperties). This can disclose different extensions of classes (especially through their subclasses). Also, properties related to classes are considered. As a last resort, *axioms* (more complex restrictions) are taken into account if they are present.

---

[9] It can be downloaded from http://nb.vse.cz/ svabo/oaei2006/#organisation.

Discussion during the Consensus building workshop also showed us the necessity of considering mappings in the neighborhood and the possibility to build mappings from some mappings that are quite certain and they have 1:1 cardinality. Reaching consensus about mappings is not an easy process, but it is achievable if people can discuss the "controversial" issues based on the facts, i.e., ontology semantics and the full context of the mappings.

The process of analysing the results of participants also addressed critical remarks to our dataset. Some ontologies contain clear mistakes in terms of hierarchy of classes, naming elements (bad English) and do not fulfil some expectations made about this dataset like richness in axioms. On the other side, these features can be so widespread in ontologies available on the present and most certainly future semantic web that it makes this dataset a truly realcase.

9 Lesson learned

From last year's lesson learned, we have applied those concerning character encoding, new evaluation measures and having a progressive test suite in the directory case. However, we must admit that not all of them have been applied, partly due to lack of time. So we reiterate those lessons that still apply with new ones, including:

A) It is now a general trend that tools for the semantic web are more robust and compliant. As a consequence, we had comments on the tests this year that concerned problems not discovered in previous years. Obviously the tools can now better handle the ontologies proposed in the tests and they return results that are more easy to handle for the evaluation. Moreover, we had more participants able to handle large scale sets.

B) Not all the systems from the last year campaign participated in the campaign of this year. Fortunately, the best system participated this year as well. It will be useful to investigate if this is a definitive trend, whether we are evaluating research prototypes or "serious" systems.

C) The benchmark test case is not discriminant enough between systems. It is still useful for evaluating the strength and weakness of algorithms but does not seems to be sufficient anymore for comparing algorithms. We will have to look into better alternatives.

D) We have had more proposals for test cases this year (we had actively looked for them). However, the difficult lesson is that proposing a test case is not enough, there is a lot of remaining work in preparing the evaluation. Fortunately, with tool improvements, it will be easier to perform the evaluation.

E) It would be interesting and certainly more realistic, to provide some random gradual degradation of the benchmark tests (5% 10% 20% 40% 60% 100% random change) instead of a general discarding of a feature. This has not been done for reason of time.

F) Last but not least, as last year we must mention that the timeline for this evaluation is far from being ideal both from the participants and the evaluators points of view. More time must be allocated to this campaign next year.

10 Future plans

Future plans for the Ontology Alignment Evaluation Initiative are certainly to go ahead and to improve the functioning of the evaluation campaign. This involves:

- Finding new real world test cases;
- Improving the tests along the lesson learned;
- Accepting continuous submissions (through validation of the results);
- Improving the measures to go beyond precision and recall (we have done this for generalized precision and recall as well as for using precision/recall graphs, and will continue with other measures);
- Drawing lessons from the new test cases and establishing general rules for consensus reference and application-oriented evaluation.

Of course, these are only suggestions that will be refined during the coming year.

11 Conclusion

The tests that have been run this year were even more complete than those of the previous years. However, more teams participated and the results tend to be better. This shows that, as expected, the field of ontology matching is getting stronger (and we hope that evaluation has been contributing to this progress).

Reading the papers of the participants should help people involved in ontology matching to find what makes these algorithms work and what could be improved.

The Ontology Alignment Evaluation Initiative will continue these tests by improving both test cases and testing methodology for being more accurate. Further information can be found at:

`http://oaei.ontologymatching.org`.

Acknowledgments

We warmly thank each participant of this contest. We know that they have worked hard for having their results ready and they provided insightful papers presenting their experience. The best way to learn about the results remains to read what follows.

We are grateful to the teams of the University of Washington and the University of Manchester for allowing us to use their ontologies of anatomy.

We also thank the other members of the Ontology Alignment Evaluation Initiative Steering committee: Lewis Hart (AT&T, USA), Tadashi Hoshiai (Fujitsu, Japan), Todd Hughes (DARPA, USA), Yannis Kalfoglou (University of Southampton, UK), John Li (Teknowledge, USA), Natasha Noy (Stanford University, USA), York Sure (University of Karlsruhe, Germany) Raphaël Troncy (CWI, Amsterdam, The Netherlands), and Petko Valtchev (Université du Québec à Montréal, Canada).

This work has been partially supported by the Knowledge Web European network of excellence (IST-2004-507482). Ondřej Šváb and Vojtěch Svátek were also partially supported by IGA VSE grants no.26/05 “Methods and tools for ontological engineering” and no.12/06 “Integration of approaches to ontological engineering: design patterns, mapping and mining”.

References

1. Ben Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proceedings of the K-Cap 2005 workshop on Integrating ontologies*, Banff (CA), 2005.
2. Paolo Avesani, Fausto Giunchiglia, and Mikalai Yatskevich. A large scale taxonomy mapping evaluation. In *Proc. 4th International Semantic Web Conference (ISWC)*, Galway (IE), 2005.
3. Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In *Proc. K-Cap 2005 workshop on Integrating ontologies*, pages 25–32, Banff (CA), 2005.
4. Jérôme Euzenat, Heiner Stuckenschmidt, and Mikalai Yatskevich. Introduction to the ontology alignment evaluation 2005. In *Proc. K-Cap 2005 workshop on Integrating ontologies*, Banff (CA), 2005.
5. Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Improving automatically created mappings using logical reasoning. In *Proc. ISWC Ontology matching workshop*, Athens (GA US), 2006. this volume.
6. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the 3rd Evaluation of Ontology-based tools workshop (EON)*, Hiroshima (JP), 2004.
7. Mikalai Yatskevich, Fausto Giunchiglia, and Paolo Avesani. A large scale dataset for the evaluation of matching systems. Technical report, University of Trento, Povo (IT), 2005.

Grenoble, Berlin, Trento, Mannheim, Prague, and Amsterdam, December 3rd, 2006

AUTOMS: Automated Ontology Mapping through Synthesis of methods

Konstantinos Kotis, Alexandros Valarakos, George Vouros

AI Lab, Information and Communication Systems Eng. Department, University of the Aegean,
Samos, 83 200, Greece
`kkot@aegean.gr, alexv@aegean.gr, georgev@aegean.gr`

Abstract. AUTOMS is a tool for the automatic alignment of domain ontologies. To ensure high precision and recall with the minimum human involvement, AUTOMS integrates several matching methods. This paper presents the tool and the results obtained for the ontologies within the framework of the OAEI 2006 contest. Particularly, the synthesis of lexical, semantic and structural matching methods, together with the exploitation of concept instances resulted in a rather high recall with space of improvement, and a quite high precision that shows the accuracy of the individual methods, as well as of their synthesis.

1 Presentation of the system

1.1 State, purpose, general statement

In this paper we present the AUTOMS tool for the automatic alignment of ontologies. The proposed tool exploits the HCONE-merge [1] ontology mapping method, which is based on "uncovering" the informal intended meaning of concepts by mapping them to WordNet senses. Furthermore, AUTOMS integrates the HCONE-merge method with an innovative lexical matcher named COCLU (COmpression-based CLUstering) [2], as well as with matching heuristics that exploit structural features of the source ontologies. The synthesis of these methods contributes towards automating the mapping process of concepts and properties of OWL ontologies, by exploiting different features of them: lexical, structural and semantic features.

The WordNet lexicon and concept instances provide additional information towards unveiling mappings in cases where features such as labels and comments are missing or in cases where names are replaced by random strings. Structure matching heuristic rules are exploited to discover mappings in situations where lexical and semantic methods do not have enough information to proceed.

AUTOMS provides mappings between concept/property pairs with high precision. However, it must be stated that it does not achieve a satisfactory recall for the experiments contacted. This suggests that further improvements are necessary both to the individual methods as well as to the sophistication of the synthesis of results.

Since the execution times for obtaining these results using OAEI contest's benchmark ontologies were quite high, a trade-off between lower time and better results was to be made.

Finally, it must be stated that AUTOMS has been improved much due to the experience gained within the OAEI contest.

1.2 Specific techniques used

The methods integrated within AUTOMS run in a particular sequence: Mappings computed by a method are being exploited by subsequent methods so as new mappings to be produced. The following paragraphs present the individual methods in the sequence of their execution.

AUTOMS is mainly based on its lexical matching method, which is applied first in the sequence of the methods employed. COCLU exploits lexical information concerning names, labels and comments of ontologies' concepts and properties, in order to compute their similarity. Although labels are considered the most important, comments and names are also examined. COCLU was originally proposed as a method for discovering typographic similarities between strings, sequences of characters over an alphabet (ASCII or UTF character set), with the aim to reveal the similarity of concepts instances' lexicalizations during ontology population [2]. It is a partition-based clustering algorithm which divides data into clusters and searches the space of possible clusters using a greedy heuristic. Each cluster is represented by a model, rather than by the collection of data assigned to it. The cluster model is realized by a corresponding Huffman tree which is incrementally constructed as the algorithm dynamically generates and updates the clusters by processing one string (instance's surface appearance) at a time. The use of a model classifies the algorithm to the conceptual or model based learning algorithms. To decide whether a new string should be added in a cluster (and therefore, that it lexicalizes the same class/property as the other strings in the cluster do) the algorithm employs a score function that measures the compactness and homogeneity of a cluster. This score function, *Cluster Code Difference (CCDiff),* is defined as the difference of the summed length of the coded string tokens that are members of the cluster, and the length of the cluster when it is updated with the candidate string. This score function groups together strings that contain the same set of frequent characters according to the model of a cluster (e.g. Pentium III and PIII). A string that lexicalizes an OWL class or property belongs in a particular cluster when its *CCDiff* is below a specific threshold and it is the smallest between the *CCDiff's* of the given string and all existing clusters. Based on our experience with COCLU, the similarity threshold (ranging in [0,1]) was set to 0.986. A new cluster is created if the candidate string cannot be assigned to any of the existing clusters. As a result, it is possible to use the algorithm even when no initial clusters are available.

Next to the computation of the lexically matching pairs is the computation of the semantic morphism (s-morphism) which is the core technique behind the HCONE-merge method. Given two ontologies, the algorithm computes a morphism between each of these two ontologies and a "hidden intermediate" ontology. This morphism is computed by the Latent Semantic Indexing (LSI) method and associates ontology concepts with WordNet senses. Latent Semantic Indexing (LSI) [4] is a vector space technique originally proposed for information retrieval and indexing. It assumes that there is an underlying latent semantic space that it estimates by means of statistical techniques using an association matrix ($n \times m$) of term-document data (WordNet senses in our case). It must be emphasized that although LSI exploits structural information of ontologies and WordNet, it ends up with semantic associations between terms. As it is specified in the HCONE-merge approach, WordNet is not considered to include any intermediate ontology, as this would be very restrictive for the specification of the original ontologies (i.e. the method would work only for those ontologies that preserve the inclusion relations among WordNet senses). Actually, it is assumed that the intermediate ontology is "hidden" and the method constructs this ontology while mapping concepts to the WordNet senses. We have used WordNet since it is a well-thought and widely available lexical resource with a large number of entries and semantic relations.

The mappings computed by the lexical and semantic matching methods are then used as input to a simple structural matching algorithm which exploits similarities in the vicinities of concepts/properties. Here, the vicinity of a concept/property includes only the subsumers and subsumees. The matching method has been implemented to improve the performance of the lexical and semantic matching methods by exploiting simple structural features. Consider the matching between two concepts c_1 and c_2 of source ontologies O_1 and O_2, respectively. The heuristic is: "if at least two neighbor concepts of c_1 have already been (lexically or semantically) mapped to two neighbor concepts of c_2 such that the mappings respect the ontology axioms of inclusion and equivalence, i.e. a sub-concept of c_1 has been mapped to a sub-concept of c_2, then c_1 and c_2 are consider to structurally match". The threshold of two (2) neighbor concepts has been considered after conducting several experiments. It should be noticed that further in the alignment process, AUTOMS uses an enhanced structure matching method that runs iteratively using the mappings of all the methods. This method expands the vicinity of concepts to include object properties as well.

The fourth method in sequence utilizes concept instances (individuals). Particularly, for the concepts that have not been determined to be similar to any other concept, AUTOMS compares their individuals, if any. For those concept pairs that have at least one matching instance, AUTOMS discovers a possible mapping. The matching of concept instances is currently based on the similarity of their local names, which is their Uniform Resource Identifier (URI).

The fifth method utilizes information about properties. For the concepts that have not been determined to be similar to any other concept, AUTOMS compares their properties, if any. For those concept pairs that have at least two matching properties, AUTOMS identifies a possible mapping. The matching of object properties is based on the similarity of their property names, as well as on the similarities of their domain and range.

The final step in the alignment process is the execution of an enhanced iterative structure matching method. This method uses the proposed matching pairs from all the previous methods in order to compute mappings based on concepts' enhanced vicinity: The enhanced vicinity of the concept includes all the concepts related to it. This method runs iteratively for 2 times, updating the list of proposed matching pairs with the pairs discovered in each iteration. It has been observed during the specific benchmark experiments that although new mappings are discovered in each iteration, there is no change to the set of mappings after the second execution. Further experimentation and investigation is needed for improving this method.

1.3 Adaptations made for the evaluation

AUTOMS is an evolving tool that integrates new methods, which are being tested in new cases. The OAEI contest provides challenging cases that require the exploitation of special features for AUTOMS (and any other tool) to perform efficiently and effectively.

Specific adaptations have been made with respect to the utilization of ‘comments’, to the existence of the ‘lang’ property, and to the use of ‘random strings’ for concept/property names.

Implementation adjustments have also been made in order to be able to run large sets of ontologies in short time. Furthermore, AUTOMS has been modified in order to be able to produce alignments in the form that OAEI contest requires. The evaluation of the results however has been performed using organizers’ Alignment API (version 2.4)

It must be noticed that for the purposes of the contest the alignment output files for ontologies 302 and 303 need to be manually fixed: The <onto2> entry (the ontology file location) value must be replaced with http://oaei.ontologymatching.org/2006/benchmarks/302/onto.rdf and http://oaei.ontologymatching.org/2006/benchmarks/303/onto.rdf, respectively.

1.4 Link to the system and parameters file

http://www.icsd.aegean.gr/incosys_old/Projects/AUTOMS/OAEI/system/automs.zip

1.5 Link to the set of provided alignments (in align format)

http://www.icsd.aegean.gr/incosys_old/Projects/AUTOMS/OAEI/results/automs.zip

2 Results

Results produced with AUTOMS for the 2006 OAEI contest are grouped and discussed below. These results were produced with a stand-alone Java version of AUTOMS on a standard Windows-based PC (2.4 GHz processor). Resulted alignments are sets of pairs of mappings, i.e. of equivalent (symb. =) concepts/properties.

2.1 Benchmark

2.1.1 Tests 101 to 104

Although these ontologies have no special features or difficulties for aligning them, AUTOMS looses in precision due to its lexical method. This is because COCLU compares first the labels of concepts/properties for their similarity, and finds 3 mappings: “number = numberOrVolume”, “collection = book”, “name = shortName”. As already pointed, although labels are considered the most important for COCLU, comments and names are also compared if labels’ similarity value is less than the specified threshold.

The semantic and structure matching methods return pairs that have already being computed by the lexical matching method, with no problems in precision. Instance and property matching methods do not contribute any mapping.

Language generalization and restriction features (103, and 104 ontologies) do not affect the results.

2.1.2 Test 201 to 210

Each case of this group of tests should be presented separately in order to thoroughly discuss the importance of each of the methods that AUTOMS integrates. We will point here to the most important issues and briefly comment each of them.

Ontologies with no names, which is the case where names have been replaced by random strings or synonyms or naming conventions or even foreign names, but with comments in place (ontologies 201, 204, 205, 206 and 207) do not cause serious problems in AUTOMS. In fact, the exploitation of the comments and the utilization of instances, as well as the mappings computed by the semantic and structure methods, result in recall that ranges from 0.66 to 1.00 and in precision that ranges from 0.94 to 0.97.

The alignment of ontologies with the above features and with no comments (ontologies 202, 209 and 210) resulted to low recall ranging from 0.10 to 0.33. The mappings were mainly contributed by the lexical method (ontologies 202, 209, and 210) and the instance matching method (ontology 202), and less by the semantic matching (ontology 209) and the enhanced structure matching method (ontology 209). Although we expected the semantic method with the use of WordNet 2.0 lexicon to unveil more mappings, it identified the pairs “Booklet = Brochure” and “Monograph = Monography”. This can be explained by the nature of most of the concept/property names (i.e. the use of compound terms or naming conventions), and by the amount/quality of information included in the vicinity of each concept/property.

Alignments of ontologies 203 and 208 have been easily produced by AUTOMS: the lack of comments did not affect much the performance. The use of labels and names, even with conventions, resulted to high precision and recall (1.00) for the ontology 203 and to precision 1.00 and recall 0.73 for the ontology 208. The exploitation of instances and the enhanced structure matching method have revealed the mappings produced for the ontology 208.

2.1.3 Test 221 to 247

Names, labels, and comments in these ontologies have no special features that may distract the alignment: These ontologies resulted from modifications in structure and the addition of instances or/and properties. That is why the recall in all test cases is 1.00. Precision is influenced by some mistakenly returned (false positive) mappings. However it does not fall under 0.87 (247 ontology), with a maximum of 1.00 in some cases (228, 233, 236 and 241 ontologies). The worst case (247 ontology) is due to false positives returned from the lexical matching algorithm and the pair "Conference = Workshop" produced by the instance matching method.

2.1.4 Test 248 to 266

These are the most difficult tests for AUTOMS since names, labels, and comments have been removed or replaced by random strings. The lexical matching method contributed only one mapping, i.e. the "lastName = lastName". Since only the instance matching method contributed in this set of alignments, the recall measure ranges from 0.10 to 0.31. The structure matching method did no contribute any mapping. The precision however is far more satisfactory, ranging from 0.82 to 1.00.

2.1.5 Test 301 to 304

Apart from ontology 304, for which the structure matching method contributed a significant number of mappings, all the mappings for these tests were computed by the lexical matching method. This fact has a negative impact to the recall measure; however precision was kept again above 0.86. The absence of concept instances, together with the fact that no semantic mappings have been computed, played a major role in the low recall measure.

2.2 Anatomy

We were not able to run this test due to the large size of ontology files.

2.4 Directory

We were able to run tests with the directory ontologies since they were given in OWL and they had a manageable size. For running this test we had to split the given ontology set in smaller sets since me experienced problems with the heap in Java (although we had used the Xmx1000M parameter). Minor adjustments to the code of AUTOMS had to be done since COCLU could not handle concept names of length 1 (e.g. ‘A’ or ‘B’) and names with numbers (e.g. ‘1990s’). The tool computed mappings between concept/property pairs, but since we had no expert mappings to evaluate our results, we can only wait for OAEI 2006 organizers’ comments.

Our observations concerning the alignments computed by AUTOMS (randomly browsing some of the 4.639) are limited to the fact that these were mainly based on the lexical matching method, secondly on the semantic matching method (e.g. “ Economics = Political_Economy ”, “ Arts = Humanities ”) and less to the structure matching method (e.g. “Regional = By_region”). Since this test was mainly addressed to discover mappings by exploiting subsumption relations, AUTOMS should have integrated a more elaborated structure matching method. Furthermore, since ontologies have no concept instances and properties, the related methods returned no mappings.

2.5 Food

We were not able to run this test due to the large size of the ontology file, as well as due to the inability of AUTOMS to import ontologies in the SKOL language.

2.6 Conference

For the purpose of this test, we run 3 separate tests, aligning the three larger ontologies (Ekaw, Iasted and OpenConf) to each of the set. Equivalences between concepts/properties were identified. The instances matching method of AUTOMS did not return any mappings, since ontologies were not populated with concept instances.

The three different sets of outputs (total of 30 alignment files in OAEI format) were used to drawn the following points:

1. A high number of mappings (14) were identified between Ekaw.owl and Conference.owl. Special features such as inverse compound names of concepts (e.g. PC_Member = Member_PC) have been tackled by the lexical matching method. Mappings such as “Document = Conference_Document” has been identified by the enhanced structure matching method. Pairs such as “Person = Human” and “Document = Article” between the Ekaw.owl and Confious.owl ontologies have been identified due to the semantic and enhanced structure matching methods, respectively. Incorrect mappings are also identified mainly due to the structure matching method (e.g. “Paper = review”) when the Ekaw.owl ontology is mapped to itself.
2. Also 14 mappings were identified between iasted.owl and sigkdd.owl ontologies. The “Delegate=Conference_Participant” mappings is identified by the enhanced structure method for the ontologies iasted.owl and Ekaw.owl. Apart from this, the rest of the mappings identified were of no particular difficulty.
3. For the openConf.owl ontology and the rest of the ontologies, the mappings that AUTOMS computed, apart from the “surname = last_name” in Confious.owl, had no specific difficulty.

4. Although several alignments have been identified between the ontologies, it seems that OAEI participants will need to spend some time prior reaching a consensus on these alignments. More important, to be able to produce an alignment between all these ontologies and finally get the reference ontology (as a result of merging them) certainly needs more time and effort. It seems that although the domain of conferencing is a generally agreed and well understood context, the different types of conferences' organizers, participants, subjects, and reviewing systems drive quite divergent ontology specifications.

Since this test is a blind test, we expect organizers' feedback.

3 General comments

The participation in the OAEI contest has been rather valuable for improving our tool. Several minor code adjustments/improvements and other methodological amendments were made in order to be able to deliver the presented precision and recall in the OAEI 2006 tests. The experience gained from the contest is keeping AUTOMS in a continuous process of improvement. In future versions of AUTOMS several problematic cases that have been discussed in this paper will be addressed.

3.1 Comments on the results

As already mentioned and also implied by the results described in this paper, AUTOMS is mainly based on the lexical matching method. The synthesized methods' performance is rather satisfactory for ontologies that use labels and comments to the specification of concepts/properties. The approach works well with naming conventions and language variations (tested for English and French). The weakness to work with concept/property names that start with numbers ('1990s') or have length of one letter ('A') has been identified and tackled. AUTOMS results are significantly based on the exploitation of concept instances (in cases where this applies). The use of structural and semantic information did not contributed as much as it was expected in the final results. However the synthesis of all these matching methods has improved the overall recall and the precision of the results.

We have collected a large set of results that AUTOMS produced for the several OAEI experiments, using variations of the individual methods and of their synthesis. AUTOMS in its simple initial version returned for the benchmark ontologies an H-mean of 0.93 and 0.63 of precision and recall, respectively. During the enrichment of AUTOMS with new and/or more advanced methods, we managed to reach the h-mean of 0.97 for precision and 0.64 of recall. The final submitted results of 0.94 (precision) and 0.67 (recall) show a good tradeoff between precision and recall that our tool can deliver for the particular benchmark ontologies.

Running AUTOMS with last year's benchmark ontologies and the Alignment API 2.4, one can observe that our tool performs better from the rest of the tools as far as the precision (0.94) is concerned, while the recall is not much lower from this of others' (0.71).

3.2 Discussions on the way to improve the proposed system

In future versions of our tool we should be experimenting with more advanced structure matching and with advanced semantic matching methods that already presented in other lines of related research [5].

Things need to be done also with the ability of AUTOMS to read and align large ontologies.

3.4 Comments on the OAEI 2006 test cases

This year OAEI contest has provided participants with harder tests and with more real ontologies. This is a good improvement on the test cases themselves. However, we think that more improvements are needed so as to avoid (at least in some of the cases) simplistic methods to dominate to more sophisticated ones: For instance, the heuristic of having only a common instance for a pair of concepts to match it seems too simplistic for contributing to the increase of methods' precision.

Furthermore, it would be more helpful to provide participants also with the reference alignments of the directory and conference tests so as to be possible for them to examine their tools, provide helpful insights to the research community and draw conclusions on real case ontologies.

3.5 Comments on the OAEI 2006 measures

Since there will always be a tradeoff between precision and recall, h-mean and/or ROC curves is a good alternative for giving participants a better view of their tool's performance. H-mean and/or ROC curves should be produced at least for each group separately since groups of tests have quite different design motivation. Also, it would be interesting to agree on the importance of precision against recall (or in the reverse order) for particular types of alignments depending on the context that they are performed. For instance, we can argue that there exist some alignments between real ontologies that are critical to being correct rather than being complete.

4 Conclusion

Our participation in the OAEI 2006 contest with the AUTOMS tool has been a significant experience. We have actually been able to identify cons and prons of our tool, and improve some points in its implementation. The organizers' feedback and the comparison with the other tools will also contribute to future improvements of the tool and of the approach in general

References

1. Kotis K., Vouros G., Stergiou K.: Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach. Elsevier's Journal of Web Semantics (JWS), vol. 4:1, (2006) pp. 60-79
2. Valarakos A., Paliouras A., Karkaletsis G., Vouros G.: A name-Matching Algorithm for supporting Ontology Enrichment. In Proceedings of SETN'04, 3rd Hellenic Conference on Artificial Intelligence, Samos, Greece (2004)
3. Miller G., Beckwith R., Fellbaum D., et al.: Five papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University (1995)
4. Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R.: Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science (1990
5. Vouros A. and Kotis K.: Extending HCONE-merge by approximating the intended interpretations of concepts iteratively. 2nd European Semantic Web Conference, Heraklion, Crete May 29 – June 1, 2005. Proceedings, Series: Lecture Notes in Computer Science, vol. 3532, Asunción Gómez-Pérez, Jérôme Euzenat (Eds.), Springer-Verlag

Appendix: Raw results

Matrix of results

#	Name	Prec.	Rec.	Time (sec)
101	Reference alignment	0.94	1.00	93
102	Irrelevat ontology	-	-	-
103	Language generalization	0.94	1.00	87
104	Language restriction	0.94	1.00	86
201	No names	0.94	0.95	86
202	No names, no comments	1.00	0.10	79
203	No comments	1.00	1.00	88
204	Naming conventions	0.94	1.00	89
205	Synonyms	0.94	0.99	89
206	Translation	0.97	0.66	84
207		0.97	0.66	84
208		1.00	0.73	82
209		1.00	0.33	81
210		1.00	0.28	78
221	No specialisation	0.94	1.00	71
222	Flatenned hierachy	0.93	1.00	81
223	Expanded hierarchy	0.89	1.00	180
224	No instance	0.94	1.00	92
225	No restrictions	0.94	1.00	64
228	No properties	1.00	1.00	44
230	Flatenned classes	0.89	1.00	80
231	Expanded classes	0.94	1.00	88
232		0.94	1.00	71
233		1.00	1.00	46
236		1.00	1.00	44
237		0.93	1.00	85
238		0.89	1.00	181
239		0.97	1.00	39
240		0.87	1.00	83
241		1.00	1.00	44
246		0.97	1.00	40
247		0.87	1.00	83
248		1.00	0.10	63
249		1.00	0.10	78
250		1.00	0.27	43
251		1.00	0.11	72
252		0.91	0.10	167
253		1.00	0.10	62
254		1.00	0.27	42
257		1.00	0.27	42
258		1.00	0.11	71
259		0.91	0.10	167
260		0.90	0.31	37
261		0.82	0.27	80
262		1.00	0.27	41
265		0.90	0.31	38
266		0.82	0.27	79
301	BibTeX/MIT	0.93	0.46	35
302	BibTeX/UMBC	1.00	0.58	24
303	Karlsruhe	0.93	0.78	141
304	INRIA	0.86	0.92	81
	H-mean	**0.94**	**0.67**	

COMA++: Results for the Ontology Alignment Contest OAEI 2006

Sabine Massmann[1], Daniel Engmann, Erhard Rahm

University of Leipzig

[1] `massmann@informatik.uni-leipzig.de`

Abstract.
This paper summarizes the OAEI Contest 2006 results for the matching tool COMA++. The study shows that a generic schema matching system can also effectively solve complex ontology matching tasks.

1 Presentation of the system

COMA++ is an extension of our previous COMA prototype [1]. It is a customizable and generic tool for matching both schemas and ontologies specified in languages such as SQL, XML Schema or OWL [2]. COMA++ offers a GUI and supports the combined use of several match algorithms as well as the reuse of previously confirmed match results [6].

The COMA++ architecture is shown in figure 1. The *Repository* persistently stores all match-related data, the *Model* and *Mapping Pools* manage all schemas, ontologies, and mappings in memory, and the *Matching Engine* performs the match operations. The GUI provides access to these components and is used to visualize models, manage the match process and mappings. The Matching Engine contains different libraries that supports many *match algorithms* and *match strategies*. The similarity results of individual matchers are maintained and aggregated within a similarity matrix per match task [1]. Match strategies implement workflows to deal with complex match tasks and enable a reuse of previous results and the decomposition of larger match tasks into smaller ones [3].

1.1 State, purpose, general statement

COMA and COMA++ have proven to be very effective for matching database and XML schemas [1, 4, 6]. The main reason for this test was to see the effectiveness of a generic matching tool for dealing with ontologies.

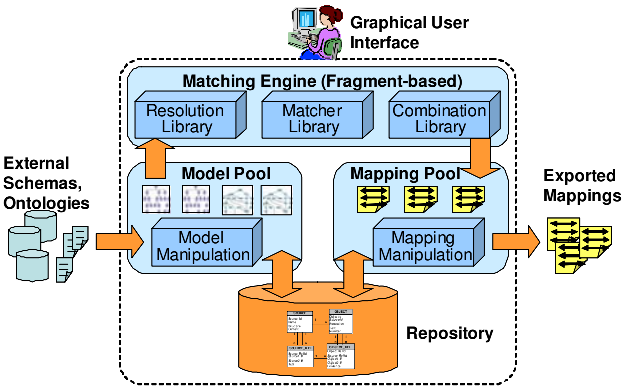


Figure 1.Architecture of COMA++

1.2 Specific techniques used

An automatic match process in COMA++ consists of several steps. In the first step the imported schemas and ontologies are transformed into a generic graph representation. The graph nodes represent schema/ontology components such as classes or properties and have attributes like name and data type. All relationships, e.g. aggregations and specializations, are uniformly represented by edges between nodes. In the next step graph nodes are matched with each other using a match strategy and matchers. There is no differentiation made between node types, so that for example classes and properties can be matched. The similarity values obtained by the individual matchers are aggregated according to a combination strategy (average, etc.). The match candidates are selected from the aggregated correspondences, e.g. based on a threshold criterion. Finally, the result mapping (RDF alignment) is generated.

In addition to the schema-based matchers we used an instance-level matcher which has recently been added to the COMA++ match library.

1.3 Adaptations made for the evaluation

In addition to the integration of an instance matcher only few changes to COMA++ were necessary to deal with specifics of the contest. As mentioned, the output mapping was translated into the predefined RDF alignment format. Furthermore the result of a matcher was ignored if it contained the same similarity value for all entities. This was a minor adaptation made because the same strategy had to be used for all tests.
Another change was the splitting of huge ontologies into several smaller ones. The results of the smaller match tasks were then merged. Another selection step was applied on the merged results to obtain the final result mapping.

To fit the rules of the contest the prototype is not using synonyms and abbreviations which can be given to the system. The specific creation of them was not allowed but would have been necessary because of the different domains.

1.4 Link to the system, parameters file and to the set of provided alignments

At the following URL .zip archives of all the contest results are available. Furthermore the system with a parameters file can be downloaded.

http://dbs.uni-leipzig.de/Research/coma_oaei.html

2 Results

The results discussed here have been calculated with five matchers: NameType, Comment, Parents, Children and Instance. For the combination of the match results the average value has been computed and a selection has been made using, e.g. a threshold. The best setting has been determined by running different configurations on the benchmark and choosing the one with the highest f-measure. The exact parameters can be found in the appendix.

2.1 Benchmark

This test is a systematic benchmark test containing 50 tests which can be used for identifying the strengths and weaknesses of an algorithm.

The overall score of COMA++ for this task (except 102) is quite good:

	Precision	Recall	F-Measure	Time
Average	0.96	0.82	0.88	7.0 sec

2.1.1 Tests 101-104

The results for tests 101, 103 and 104 are perfect because the classes and properties have the same names, comments and instances. The language restriction and generalization have no influence.

The alignment for the irrelevant ontology 102 contains a few false matches that have similar names, e.g. “year – yearValue”. There are no matches expected for this test, thus precision and recall automatically are 0.0, so we left this value out at the average calculation.

	Precision	Recall	F-Measure	Time
Average	1.00	1.00	1.00	15.4 sec

2.1.2 Tests 201-247

The results of these tests differ depending on the given information because the chosen strategy uses names, data types, comments, structure and instance. If one or more of these information is missing only the remaining information can be used.

For the tasks 202, 209 and 210 the names and the comments differ so these information can't be used and the results have a lower recall.

For all other tests of this group the names, the comments or both contain useful information so the results are quite good.

The tests 221-247 even have the same names and comments, whereas the structure is different. Instances are similar but some ontologies don't contain them. The given information is enough to reach very good results.

	Precision	Recall	F-Measure	Time
Average	0.98	0.95	0.97	8.1 sec

2.1.3 Tests 248-266

In these tests the names have been substituted with random strings and there are no comments. The algorithm can thus only use the hierarchy and the instances, if given. Not for every class and property instances exist, so that information just helps to find corresponding entities. The results for these tests are therefore satisfactory.

	Precision	Recall	F-Measure	Time
Average	0.89	0.51	0.65	4.2 sec

2.1.4 Tests 301-304 (Real Ontologies)

The real-world ontologies have been a more difficult task for COMA++ because the ontologies are quite different compared with the 101 ontology. Three out of the four ontologies don't contain instances – only 304 does. 302 and 303 don't use comments, the structure is quite different and the names are often dissimilar, which the prototype could not find because the contest disallowed us to use auxiliary information.

	Precision	Recall	F-Measure	Time
Average	0.84	0.69	0.76	3.6 sec

2.2 Anatomy

For the anatomy task two large ontologies had to be aligned. Because of the huge size the matching task had to be splitted by our system into smaller ones. The part results were merged and then a variety has been selected. The selection was necessary because with the splitted matching more false matches have been found.

Another difficulty has been the fact that in the FMA ontology the id of classes look like "frame_92794" and "frame_51746" and the real information is in the label. Whereas the OpenGALEN ontology has meaningful ids and uses rarely labels. These labels or ids are made up of a lot of tokens and sometimes they differ only in a few letters, e.g. "fifth" instead of "first". Therefore we expect that more false positives will be found than in the benchmark test.

2.3 Directory

For this test we matched 4640 pairs of ontologies.
To find out more about the quality of our strategy and that kind of test we also matched the 2265 ontology pairs of the contest 2005. We reached a recall around 0.32 what is as good as the best participants. Looking at the missing correspondences we couldn't find any similarity of the names, e.g., "7/source.owl#Academic_Departments" and "7/target.owl#United_Kingdom" and no comments or instances existed. That's why we couldn't figure out a way to improve our system.

2.4 Food

The food ontologies uses the different format SKOS. We transformed the given SKOS files into OWL format to be able to match them. These ontologies are quite large so the match process has to be splitted as well as in the anatomy test.

2.5 Conference

This task contains 10 ontologies that deal with conference organisation. The calculation of alignments between each of them was no problem because of the smaller size.

3 General comments

3.1 Comments on the results

Given that COMA and COMA++ were not specifically designed for matching ontologies and we invested only a small amount of time for the contest the overall results are surprisingly good. The new instance matcher proved to be effective especially for the tests where useful information was only provided by instance values.

The used parameters were selected for the whole set of tests. For individual match tasks better results than reported can be obtained by using tailored configuration parameters. Another point is that domain-specific abbreviations, synonyms and previous match results could not be utilized in order to conform with the contest rules.

3.2 Discussions on the way to improve the proposed system

The use of auxiliary information that is conforming to the rules, e.g. WordNet or UMLS, could improve the recall results. The addition of ontology-oriented matchers and the distinction between node and relationship types could also be helpful.

3.3 Comments on the OAEI 2006 procedure

This is our first participation in this Ontology Alignment Contest. Since we are not involved in the contest preparation we had no prior knowledge of most tasks and the regulations. We thus had comparatively little time (about 2 months) to deal with the details of six test series and technical problems caused by unknown formats and large files. Furthermore, we had to adapt the system to the contest rules and try to find the best strategy and configuration.

4 Conclusion

The presented contest results show that COMA++ is not only effective for schema matching but also for ontology matching. This underlines the viability of generic approaches for complex metadata management problems.

References

1. Do, H.H., E. Rahm: COMA - A System for Flexible Combination of Schema Matching Approach. Proc. Intl. Conf. Very Large Databases (VLDB), 2002
2. Aumüller, D., H.H. Do, S. Massmann, E. Rahm: Schema and Ontology Matching with COMA++ (Software Demonstration). Proc. 24. ACM SIGMOD Intl. Conf. Management of Data, 2005
3. Rahm, E., H.H. Do, S. Massmann: Matching Large XML Schemas. ACM SIGMOD Record 33(4), 2004
4. Do, H.H., S. Melnik, S., E. Rahm: Comparison of Schema Matching Evaluations. Proc. 2. Intl. Workshop Web and Databases, LNCS 2593, Springer Verlag, 2003
5. Rahm, E., P.A. Bernstein: A Survey of Approaches to Automatic Schema Matching. VLDB Journal, 10(4), 2001
6. Hong-Hai Do. Schema Matching and Mapping-based Data Integration, Dissertation, Department of Computer Science, Universität Leipzig, Germany, 2006

Appendix: Raw results

The following benchmark results have been computed with the following parameters:

- Strategie: NoContext
- Matcher: NameType, Comment, Instance, Parents, Children
- Combination: Average
- Selection: N=0, Delta=0.0001, Threshold=0.13; Direction=Both

The tests were run on a PC running Windows XP with an Intel Pentium 4 2.4 GHz processor and 512 MB memory.

Matrix of results

#	Name	Prec.	Rec.	Time (sec)
101	Reference alignment	1.00	1.00	15.9
102	Irrelevat ontology	0.00	0.00	5.8
103	Language generalization	1.00	1.00	16.5
104	Language restriction	1.00	1.00	13.7
201	No names	1.00	1.00	14.0
202	No names, no comments	0.90	0.68	12.1
203	No comments	1.00	1.00	12.5
204	Naming conventions	1.00	1.00	14.5
205	Synonyms	1.00	0.98	13.6
206	Translation	1.00	0.98	14.0
207		1.00	0.98	13.2
208		0.99	0.98	11.8
209		0.96	0.78	12.4
210		0.98	0.85	13.4
221	No specialisation	1.00	1.00	4.4
222	Flatenned hierachy	1.00	1.00	5.1
223	Expanded hierarchy	1.00	1.00	6.5
224	No instance	1.00	1.00	4.2
225	No restrictions	1.00	1.00	12.9
228	No properties	0.94	0.94	3.0
230	Flatenned classes	0.99	1.00	11.8
232		1.00	0.99	7.7
233		0.94	0.94	3.0
236		0.94	0.94	3.5
237		1.00	1.00	4.3
238		0.99	0.99	4.9
239		0.90	0.93	3.3
240		0.79	0.91	3.6
241		0.94	0.94	2.8
246		0.90	0.93	2.7
247		0.77	0.91	4.2

248		0.91	0.52	4.1
249		0.89	0.68	12.6
250		0.93	0.42	3.3
251		0.83	0.57	4.5
252		0.90	0.57	4.9
253		0.91	0.52	3.6
254		1.00	0.27	2.9
257		0.93	0.42	2.6
258		0.84	0.58	4.7
259		0.90	0.57	5.3
260		0.86	0.41	2.5
261		0.92	0.33	3.2
262		1.00	0.27	2.7
265		0.86	0.41	2.6
266		0.92	0.33	2.9
301	BibTeX/MIT	0.97	0.64	3.8
302	BibTeX/UMBC	0.78	0.44	2.4
303	Karlsruhe	0.62	0.65	4.0
304	INRIA	0.96	0.91	4.3

DSSim-ontology mapping with uncertainty

Miklos Nagy, Maria Vargas-Vera, Enrico Motta

Knowledge Media Institute (Kmi)
The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
mn2336@student.open.ac.uk;{m.vargas-vera,e.motta}@open.ac.uk

Abstract. This paper introduces an ontology mapping system that is used with a multi agent ontology mapping framework in the context of question answering. Our mapping algorithm incorporates the Dempster Shafer theory of evidence into the mapping process in order to improve the correctness of the mapping. Our main objective was to assess how applying the belief function can improve correctness of the ontology mapping through combining the similarities which were originally created by both syntactic and semantic similarity algorithms. We carried out experiments with the data sets of the Ontology Alignment Evaluation Initiative 2006 which served as a test bed to assess both the strong and weak points of our system. The experiments confirm that our algorithm performs well with both concept and property names.

1. Presentation of the system

1.1 State, purpose, general statement

In the context of the Semantic Web, AQUA [1,2] an ontology based question answering system offers the possibility to answer user queries from heterogeneous data sources described by their own domain specific ontologies. In order to produce coherent answer to the users' query in this distributed environment the AQUA system need to create ontology mappings between both the concepts and properties of the different domains and the query terms posed by the user. However, in the context of question answering like the AQUA system the dynamic nature of the source information (e.g. web enabled databases) does not always make it possible to create ontology mapping a-priory by the help of a domain expert, but mappings need to be created on the fly. Considering the dynamic nature of this environment an important aspect is how the incomplete and uncertain results of the different similarity algorithms can be interpreted during the mapping process. We believe that proper utilization of uncertainty can considerably improve the mapping precision. However, uncertain data handling and combining uncertain data obtained from different sources in general is computationally expensive operation therefore we use multi agent architecture to address performance related issues.

1.2 Specific techniques used

Creating the particular ontology mappings is an iterative process where ideally the users are involved in the loop as well. In a real case scenario the users pose different questions that contain both concepts and properties of a particular domain. This information then can be used to query the different ontologies, create mapping between its concepts and properties that can be used to answer the particular query. For the Ontology Alignment Contest we have implemented an iterative closed loop which creates the mapping without any human interaction and works as follows:

1. We take a concept (or property) from ontology 1 and consider (refer to it from now) it as the query fragment that would normally be posed by a user. From the query fragment we build up a graph which contains the close context of the query fragment such as the concept and its properties.
2. We take syntactically similar concepts and properties and its synonyms to the query graph from ontology 2 and build a graph that contains both concepts (properties) and its synonyms.

3. Different similarity algorithms (considered as different experts in evidence theory) are used to assess quantitative similarity values (converted into belief mass function) between the nodes of the query and ontology fragment which is considered as an uncertain and objective assessment. Then the information produced by the different algorithms is combined using the Dempster's rule of combination.
4. Based on the combined evidences we assess semantic similarity between the query and ontology graph fragment structures and select those in which we calculate the highest belief function.
5. The selected concepts are added into the alignment.

The overview of the mapping process is depicted on figure 1.

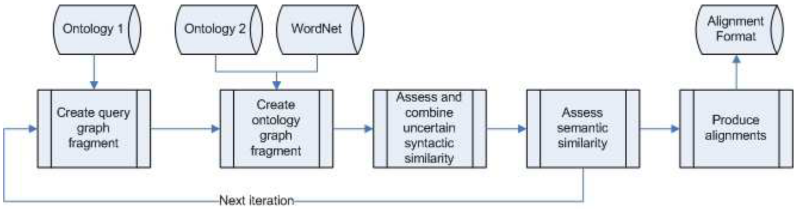


Fig. 1. The iterative mapping process

In order to avoid a complex graph of relationships in the query and the ontology fragments we need to define a reasonable limit on the number of synonyms, which are extracted from the WordNet. To define such a limit is also desirable when we carry out the belief combination since all extracted terms represent a variable where each similarity value needs to be combined with the Dempster's rule of combination. The combination rule implies that the problem space increases exponentially with the number of variables therefore the proper definition of this limit can considerably affect the scalability of our system.

1.2.1 Syntactic similarity

To assess syntactic similarity between ontology entities we use different string-based techniques to match names and name descriptions. These distance functions map a pair of strings to a real number, which indicates a qualitative similarity between the strings. To achieve more reliable assessment we combine different string matching techniques such as edit distance like functions e.g. Monger-Elkan [3] to the token based distance functions e.g. Jaccard [4] similarity. To combine different similarity measures we use Dempster's rule of combination. Several reasonable similarity measures exist however, each being appropriate to certain situations. To maximize our system's accuracy we employ a variety of similarity measures. At this stage of the similarity mapping our algorithm takes one entity from Ontology 1 and tries to find similar entity in extended query. The similarity mapping process is carried out on the following entities:

- Concept-name similarity
- Property name and set similarity

The use of string distances described here is the first step towards identifying matching entities between query and the ontology or between ontologies with little prior knowledge, or ill structured data. However, string similarity alone is not sufficient to capture the subtle differences between classes with similar names but different meanings. So we work with WordNet in order to exploit synonymy at the lexical-level. Once our query sting is extended with lexically synonym entities we calculate the string similarity measures between the query and the ontologies. In order to increase the correctness of our similarity measures the obtained similarity coefficients need to be combined. Establishing this combination method was our primary objective that had been included into the system. Further once the combined similarities have been calculated we developed a simple methodology to derive the belief mass function that is the fundamental property of Demster-Shafer framework.

1.2.2 Semantic similarity

For semantic similarity between concept, relations and the properties we use graph-based techniques. We take the extended query and the ontology input as labeled graphs. The semantic matching is viewed as graph-like structures containing terms and their inter-relationships. The similarity comparison between a pair of nodes from two ontologies is based on the analysis of their positions within the graphs. Our assumption is that if two nodes from two ontologies are similar, their neighbours might also be somehow similar. We consider semantic similarity between nodes of the graphs based on similarity of leaf nodes. That is, two non-leaf schema elements are semantically similar if their leaf sets are highly similar, even if their immediate children are not. The main reason why semantic heterogeneity occurs in the different ontology structures is because different institutions develop their data sets individually, which as a result contain many overlapping concepts. Assessing the above-mentioned similarities in our system we adapted and extended the SimilarityBase and SimilarityTop algorithms [5] used in the current AQUA system for multiple ontologies. Our aim is that the similarity algorithms (experts in terms of evidence theory) would mimic the way a human designer would describe a domain based on a well-established dictionary. What also needs to be considered when the two graph structures are obtained from both the user query fragment and the representation of the subset of the source ontology is that there can be a generalization or specialization of a specific concepts present in the graph which was obtained from the local source and this needs to be handled correctly. In our system we adapted and extended the before mentioned SimilarityBase and SimilarityTop algorithms, which has been proved effective in the current AQUA system for multiple ontologies.

1.2.3 Uncertainty

In our system we use the Dempster-Shafer theory of evidence [6], which provides a mechanism for modeling and reasoning uncertain information in a numerical way particularly when it is not possible to assign a belief to a single element of a set of values. Consequently the theory allows the user to represent uncertainty for knowledge representation, because the interval between support and plausibility can be easily assessed for a set of hypothesizes. Missing data also could be modeled by Dempster-Shafer approach and additionally evidences from two or more sources can be combined using Dempster's rule of combination. The combined support, plausibility, disbelief and uncertainty can each be separately evaluated. The main advantage of the Dempster-Shafer theory over the classical probabilistic theories is the evidence of different levels of abstraction can be represented in a way, which allows clear distinction to be made between uncertainty and ignorance. Further advantage is that the theory provides a method for combining the effect of different learned evidences to establish a new belief by using Dempster's combination rule. The following elements have been used in our system in order to model uncertainty:
Belief mass function (m): is a finite amount of support assigned to the subset of Θ. It represents the strength of some evidence and

$$\sum_{A\subseteq\Theta} m(A) = 1 \qquad \textbf{(1)}$$

where m(A) is our exact belief in a proposition represented by A. The similarity algorithms itself produce these assignment based on the above mentioned (see in section similarity) similarities. As an example consider the query fragment that contains the concept "book". Based on the WordNet we identify that the concept "volume" is one synonym of the "book" so after similarity assessment our variables will have the following belief mass value:

- $m(\text{Ontology1}_{\text{book}}, \text{Ontology2}_{\text{volume}}) = 0.89$
- $m(\text{Ontology1}_{\text{book}}, \text{Ontology2}_{\text{book}}) = 1.0$

In practice we would assess up to 8 synonym similarities with different algorithms (considered as experts) which can be combined based on the combination rule in order to create a more reliable mapping. Once the combined belief mass functions have been assigned the following additional measures can be derived from the available information.
Belief: amount of justified support to A that is the lower probability function of Dempster, which accounts for all evidence E_k that supports the given proposition A.

$$belief_i(A) = \sum_{E_k \subseteq A} m_i(E_k) \qquad \textbf{(2)}$$

An important aspect of the mapping is how one can make a decision over how different similarity measures can be combined and which nodes should be retained as best possible candidates for the match. To combine the qualitative similarity measures that have been converted into belief mass functions we use the Dempster's rule of combination and we retain the node which belief function has the highest value.

Dempster's rule of combination:
Suppose we have two mass functions $m_i(E_k)$ and $m_j(E_k')$ and we want to combine them into a global $m_{ij}(A)$. Following Dempster's combination rule

$$m_{ij}(A) = m_i \oplus m_j = \sum_{E_k \cap E_{k'}} m_i(E_k) * m_j(E_{k'}) \qquad \textbf{(3)}$$

1.2.4 Trust and conflicting beliefs

Based on our experiments with the benchmarks we have investigated why in some cases the belief combination produced incorrect result even thought before the combination a correct mapping could have been derived for the particular case based on individual beliefs. The problem occurs when the different agents' similarity assessment produces conflicting beliefs over the correctness of a particular mapping. A conflict between two beliefs in DS theory can be interpreted qualitatively as one source strongly supports one hypothesis and the other strongly supports another hypothesis, and the two hypotheses are not compatible. In this scenario applying Dempster's combination rule to conflicting beliefs can lead to an almost impossible choice with a very low degree of belief which due to the normalisation will result in the most possible outcome with a very high degree of belief [7, 8]. This combination rule strongly emphasizes the agreement between multiple sources and ignores all the conflicting evidence through a normalization factor. Imagine the following scenario where Ω frame of discernment has three elements {e_1,e_2,e_3} and the assigned belief masses on the correctness of the particular mappings are as described on table 1.

	Before normalisation	After normalisation
Agent 1	$m_1(e_1) = 0; m_2(e_2) = 0.01; m_3(e_3) = 0$	$m_1(e_1) = 0; m_2(e_2) = 1; m_3(e_3) = 0$
Agent 2	$m_1(e_1) = 0.74; m_2(e_2) = 0.35; m_3(e_3) = 0.24$	$m_1(e_1) = 0.55; m_2(e_2) = 0.26; m_3(e_3) = 0.19$
Agent 3	$m_1(e_1) = 0.69; m_2(e_2) = 0.3; m_3(e_3) = 0.21$	$m_1(e_1) = 0.57; m_2(e_2) = 0.25; m_3(e_3) = 0.18$

Table 1. Conflicting belief masses

In this scenario the belief of "Agent 1" is in conflict with the other agents' belief and due to the normalization of the hypothesis set a week possibility is transformed into strong support which would result in an incorrect mapping. In our ontology mapping framework the belief functions are considered as a method to model an agent's beliefs, therefore the belief function defined by an agent can also be viewed as a way of expressing the agent's preferences over choices, with respect to masses assigned to different hypotheses. The larger the mass assigned to a hypothesis is the more preferred the hypothesis will be. In this context the problem is how do we handle the agent's conflicting individual preferences that need to be aggregated in order to form a collective preference. We have utilized the degree of trust based on reputation model [9] between the individual agents' belief over the correctness of the mapping. In our scenario the reputation model is particularly appealing because it can be defined as the collective opinion or view about the mapping where this view can be mainly be derived from an aggregation of individual preferences. In our ontology mapping framework we assess trust between the agent's beliefs and determine which agent's belief cannot be trusted ignoring the one which contradicts with the majority of the beliefs which are similar to each other.

1.3 Adaptations made for the evaluation

Our mapping algorithm which is originally based on multi agent architecture has been re-implemented as a standalone mapping process which uses the common WordNet dictionary which is considered more general knowledge then originally we assume in or architecture. Originally our mapping process receives query fragments from the AQUA system where the query fragments contain several concept names and their properties. For the evaluation we modified our mapping process so we consider the individual concept or property names as query fragments which contain less information about the possible mapping then the query fragments that we originally receive from the AQUA system.

1.4 Link to the system and parameters file

http://kmi.open.ac.uk/people/miklos/OAEI2006/DSSemanticSimilarity.zip

1.5 Link to the set of provided alignments (in align format)

http://kmi.open.ac.uk/people/miklos/OAEI2006/benchmarks.zip

2. Results

All the tests have been carried out on a commercially available notebook with windows operating system. The mapping algorithm has been implemented in Java and been integrated with the Alignment api. The comments are made on the tests that have been grouped as follows:

2.1 Tests 101-104

The ontologies include (see figure 2) the reference alignment and irrelevant ontology a language generalization and a language restriction. Our results (see result matrix) show that our mapping algorithm creates the mapping with high precision for this tests.

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
101	1	0.98	0	0.99	0.98
102	0	NaN	1	NaN	NaN

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
103	1	0.98	0	0.99	0.98
104	1	0.98	0	0.99	0.98
H-mean	1	0.98	0	0.99	0.98

Fig. 2. Results from test 101-104

2.2 Tests 201-210

The ontology 201 that does not contain names and 202 which neither contain names nor comments were not mapped at all by our algorithm. Our algorithms considers only class and property IDs as identified by the “rdf:ID” tag therefore the only information that can be used to create these mappings the “rdfs:comment” but our algorithm does not make use of it. Ontologies 203 and 204 are without comments and certain naming conventions were also mapped with high precision by our algorithm. Ontology 205 which contains synonyms were mapped with high precision but with really weak recall what can be explained by the fact that our algorithm looks for WordNet synonyms based on the full terms from the ontologies so e.g. MastersThesis or MScThesis as one word does not have WordNet synonym but MSc and Thesis separately do. Ontologies 206 to 210 are the French translations of the original ontology and since our algorithm does not look at the comments therefore our mapping has a low recall rate. The results of the mappings for this group are depicted on figure 3.

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
201	NaN	0	NaN	NaN	NaN
202	NaN	0	NaN	NaN	NaN
203	1	0.98	0	0.99	0.98
204	0.99	0.68	0.01	0.8	0.67
205	0.88	0.23	0.12	0.36	0.2

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
206	0.91	0.21	0.05	0.34	0.2
207	0.91	0.21	0.09	0.34	0.19
208	0.99	0.68	0.01	0.8	0.67
209	0.88	0.23	0.12	0.36	0.2
210	0.91	0.21	0.09	0.34	0.19
H-mean	0.95	0.34	0.04	0.5	0.33

Fig. 3. Results from test 201-210

2.3 Tests 221-247

Ontologies from 221 to 247 (see figure 4) contain no specialization, flatenned hierarchy, expanded hierarchy, no instance, no restrictions, no datatypes, unit difference, no properties, class vs instances, flattened classes and expanded classes have been mapped with a very high recall and precision rate. We can conclude that on this group of tests our algorithm performs well which can be contributed to the fact that we carry out both syntactic and semantic similarity assessment.

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
221	1	0.98	0	0.99	0.98
222	1	0.98	0	0.99	0.98
223	1	0.98	0	0.99	0.98
224	1	0.98	0	0.99	0.98
225	1	0.98	0	0.99	0.98
228	1	1	0	1	1
230	0.99	0.97	0.01	0.98	0.96
231	1	0.98	0	0.99	0.98
232	1	0.98	0	0.99	0.98

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
233	1	1	0	1	1
236	1	1	0	1	1
237	1	0.98	0.01	0.98	0.97
238	1	0.98	0	0.99	0.98
239	1	1	0.03	0.98	0.97
240	1	1	0	1	1
241	1	1	0	1	1
246	1	1	0.03	0.98	0.97
247	1	1	0	1	1
H-mean	0.99	0.98	0	0.99	0.98

Fig. 4. Results from test 221-247

2.4 Tests 248-266

Again since our algorithm considers only class and property IDs as identified by the “rdf:ID” tag therefore these tests have not produced any mapping. In a future implementation we will considerer labels. Then, our similarity algorithm will be able to handle effectively these cases.

2.5 Tests 301-304

For the real word ontologies (see figure 5) our algorithm produced relatively good mappings with good recall and high precision. We believe that the real word ontlogies and the reference ontology were not so different semantically in terms of concept and property hierarchies or structure so the syntactic similarity was dominated the results.

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
301	0.87	0.79	0.13	0.83	0.67
302	0.93	0.58	0.07	0.72	0.54

algo	DSSim				
test	Prec.	Rec.	Fall.	FMeas.	Over.
303	0.84	0.78	0.16	0.81	0.63
304	0.94	0.89	0.06	0.92	0.84
H-mean	0.9	0.78	0.1	0.83	0.69

Fig. 5. Results from test 301-304

3. General comments

3.1 Comments on the results (strength and weaknesses)

We consider the results successful when we reach a high precision rate since our main objective is to increase ontology mapping precision with incorporating uncertainty into the mapping process. Most of the benchmark tests proved that when different similarity assessments have to be combined handling uncertainty can lead to a high precision rate which is a definite strength of our system. Another strength of our system is that the produced mappings are not very dependent on the structure and hierarchy of the concepts and properties in the ontology (see tests 221-247). Since the multi agent architecture has been replaced with the single process the execution time has increased considerably. Additionally the agents's "specific knowledge" has been replaced with the general WordNet synonyms that negatively influenced the system. Further our algorithm always considers the ID tag in the ontologies therefore any additional information like comments or the language element is omitted. Not considering the language element can be considered as a weakness. However, we believe that comments in ontologies can work well when the ontologies originate from a well controlled environment with strong academic background like universities or research institutions. From the another side if we consider the nature of the semantic web where any private company can place its ontology to the web to support its own web enabled data it can lead to really different comments even for the same concepts or properties.

3.2 Discussions on the way to improve the proposed system

Based on the results we have identified the following improvement possibilities that can further improve our system:

1. We need to split it up the concept and property IDs in the ontologies which are the combination of two or more different terms e.g. MScThesis into unique terms and the WordNet synonyms can be retrieved on the combination of the separated terms. This can lead to a definite improvement of recall number of the particular mapping.
2. Wherever possible or present considering the language tag as primary information. It is important that we create mapping based on the same language. Failing to do so can lead to incorrect mappings that cannot be detected based on qualitative measures.

3.3 Comments on the OAEI procedure

The OAEI procedure and the provided alignment api works well for the benchmarks. However we experienced difficulties with the anatomy ontology. We have tried on several computers but we have always got **OutOfMemoryError** due to the large size of the FMA ontology. Our investigation showed that when the GroupAlign class of the alignment api parses the source and target ontologies into a **org.semanticweb.owl.model.OWLOntology** object the memory usage of the JVM process increases to nearly 1GB. Once the similarity mapping process starts, any manipulation of the original ontology object leads to OutOfMemoryError and causes the process to stop. We have also tried to increase the stack size of the JVM but it did not solve the problem.

3.4 Comments on the OAEI test cases

We have found that most of the benchmark tests can be used effectively to test various aspects of an ontology mapping system since it provides both real word and generated/modified ontologies. The ontologies in the benchmark are conceived in a way that allows anyone to clearly identify system strengths and weaknesses which is an important advantage when future improvements have to be identified. However, our system did not perform as well as we first expected probably due to the fact that most of the classes and properties in the ontologies are organized in a rather flat hierarchy so in our system the semantic similarity component did not influence the overall mappings considerably. Unfortunately, we could not make use of a large group of tests (248-266) since our system does not consider individuals or instances of the classes. Concerning the anatomy data sets we planned to produce alignment as well however, we were unable to successfully run the process using the alignment api due to the reasons described in the section 3.3. The external and blind evaluations are certainly valuable exercises however we plan to utilize them in the future due to technical limitations of our system.

3.4 Comments on the OAEI measures

For our system the precision measure was the most important of all because this gives us the possibility to draw constructive conclusions on how the uncertainty handling can influence the precision of the system. The additional measures like recall and fallout can be used effectively for identifying where do we need to make further improvements in our system.

3.5 Proposed new measures

Besides the traditional measures it would be useful as well to introduce a measure that expresses the difficulty to create the particular mapping. E.g. there is a considerable difference in the level of difficulty between creating mapping with the reference ontology itself (101 to 101) and real word ontology (101 to 304). This measure then could be used to assess the how the particular system can handle mappings that involves complex comparison operations.

4. Conclusions

The increasing popularity of the Semantic Web poses new challenges for ontology mapping. If we accept that mapping ontologies can provide a better knowledge management of the heterogeneous sources on the Semantic Web, then issues of inconsistency and incompleteness need to be addressed. Therefore ontology mapping systems that operate in this environment should have the appropriate mechanisms to cope with these issues. In this complex environment different scientific disciplines need to be utilized together to achieve better results for answering user queries within an acceptable response times. We think that in our implementation we have made an encouraging step towards a theoretical solution but the different key system components such as similarity measure or the scalability of uncertainty handling part needs to be investigated further. In our future research we will investigate how different optimalisation methods for belief combination can be adapted and applied in our scenario with a dynamic multi agent environment where each agent has partial knowledge of the domain. Participating in the Ontology Alignment Evaluation Initiative is an excellent opportunity to test and compare our system with other solutions and helped a great deal identifying the future possibilities that needs to be investigated further.

References

[1] Vargas-Vera, M. and Motta, E. 2004. AQUA - Ontology-based Question Answering System. In Proceedings of Third International Mexican Conference on Artificial Intelligence (MICAI-2004), Mexico City, Mexico.

[2] Vargas-Vera, M.; Motta, E.; and Domingue, J. 2003. AQUA: An Ontology-Driven Question Answering System. AAAI Spring Symposium, New Directions in Question Answering, Stanford, USA.:AAAI Press.

[3] Monge, A. E.; Elkan, C. P. 1996. The field-matching problem: algorithm and applications. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, US.

[4] Cohen, W.; Ravikumar, P.; and Fienberg, S. 2003. A Comparison of String Distance Metrics for Name-Matching Tasks, In Proceedings of Information Integration on the Web (IIWeb 2003), Accapulco, Mexico.

[5] Vargas-Vera, M.; and Motta, E. 2004. An Ontology-driven Similarity Algorithm. KMI-TR-151, Knowledge Media Institute, The Open University, UK.

[6] Shafer, G. 1976. A Mathematical Theory of Evidence.: Princeton University Press.

[7] Smets, Ph. 1988. Belief functions, In Non-Standard Logics for Automated Reasoning: Academic Press, pp. 253–286.

[8] Zadeh, L. 1986. A simple view of the Dempster–Shafer theory of evidence and its implication for the rule of combination, AI Magazine 7, 85–90.

[9] Ramchurn, S. D., Huynh, T. D. and Jennings, N. R. 2004. The Knowledge Engineering Review, Volume 19: 1-25.

Appendix

Matrix format

algo	**DSSim**				
test	Prec.	Rec.	Fall.	FMeas.	Over.
101	1	0.98	0	0.99	0.98
102	0	NaN	1	NaN	NaN
103	1	0.98	0	0.99	0.98
104	1	0.98	0	0.99	0.98
201	NaN	0	NaN	NaN	NaN
202	NaN	0	NaN	NaN	NaN
203	1	0.98	0	0.99	0.98
204	0.99	0.68	0.01	0.8	0.67
205	0.88	0.23	0.12	0.36	0.2
206	0.91	0.21	0.05	0.34	0.2
207	0.91	0.21	0.09	0.34	0.19
208	0.99	0.68	0.01	0.8	0.67
209	0.88	0.23	0.12	0.36	0.2
210	0.91	0.21	0.09	0.34	0.19
221	1	0.98	0	0.99	0.98
222	1	0.98	0	0.99	0.98
223	1	0.98	0	0.99	0.98
224	1	0.98	0	0.99	0.98
225	1	0.98	0	0.99	0.98
228	1	1	0	1	1
230	0.99	0.97	0.01	0.98	0.96
231	1	0.98	0	0.99	0.98
232	1	0.98	0	0.99	0.98
233	1	1	0	1	1
236	1	1	0	1	1
237	1	0.98	0.01	0.98	0.97

algo	**DSSim**				
test	Prec.	Rec.	Fall.	FMeas.	Over.
238	1	0.98	0	0.99	0.98
239	1	1	0.03	0.98	0.97
240	1	1	0	1	1
241	1	1	0	1	1
246	1	1	0.03	0.98	0.97
247	1	1	0	1	1
248	NaN	0	NaN	NaN	NaN
249	NaN	0	NaN	NaN	NaN
250	NaN	0	NaN	NaN	NaN
251	NaN	0	NaN	NaN	NaN
252	NaN	0	NaN	NaN	NaN
253	NaN	0	NaN	NaN	NaN
254	NaN	0	NaN	NaN	NaN
257	NaN	0	NaN	NaN	NaN
258	NaN	0	NaN	NaN	NaN
259	NaN	0	NaN	NaN	NaN
260	NaN	0	NaN	NaN	NaN
261	NaN	0	NaN	NaN	NaN
262	NaN	0	NaN	NaN	NaN
265	NaN	0	NaN	NaN	NaN
266	NaN	0	NaN	NaN	NaN
301	0.87	0.79	0.13	0.83	0.67
302	0.93	0.58	0.07	0.72	0.54
303	0.84	0.78	0.16	0.81	0.63
304	0.94	0.89	0.06	0.92	0.84
H-mean	0.98	0.55	0.02	0.7	0.53

The Results of Falcon-AO in the OAEI 2006 Campaign

Wei Hu, Gong Cheng, Dongdong Zheng, Xinyu Zhong, and Yuzhong Qu

School of Computer Science and Engineering, Southeast University,
Nanjing 210096, P. R. China
{`whu, gcheng, ddzheng, xyzhong, yzqu`}`@seu.edu.cn`

Abstract. In this paper, we briefly introduce the architecture of Falcon-AO (version 0.6) and highlight two major improvements in the current version. Falcon-AO successfully completes all the five alignment tasks in the OAEI 2006 campaign: `benchmark`, `anatomy`, `directory`, `food`, and `conference`, and some preliminary results are also reported in this paper. In the end, we present some comments about our results and lessons learnt from the campaign towards building a comprehensive ontology alignment system.

1 Presentation of the System

As an infrastructure for the Semantic Web applications, Falcon is a vision of our research group. It desires for providing fantastic technologies for finding, aligning and learning ontologies, and ultimately for capturing knowledge by an ontology-driven approach. It is still under development in our group. As a prominent component of Falcon, Falcon-AO is an automatic tool for aligning ontologies, which is dedicated to aligning the Web ontologies expressed in OWL Lite/DL. To date, Falcon-AO is continually being improved and elaborated, and currently the latest version is 0.6.

1.1 State, Purpose, General Statement

Falcon-AO is an automatic ontology alignment tool. There are three elementary matchers implemented in the current version: V-Doc [4], I-Sub [5], and GMO [1]. In addition, an ontology partitioner, PBM [2], is integrated into Falcon-AO to cope with large-scale ontologies. In order to coordinate all the elementary matchers with high quality, we devise a novel central controller, which is based on the observation of the linguistic comparability as well as the structural comparability. The architecture of Falcon-AO (version 0.6) is illustrated in Fig. 1.

Compared with our previous prototype (version 0.3) [3], Falcon-AO (version 0.6) is extended mainly in two aspects. One is the integration of PBM. The other is the design of the central controller. The details about the two improvements are presented in the next subsection. Besides, it is worthy of noting that we also refine the implementation of the elementary matchers to save the runtime of matching process.

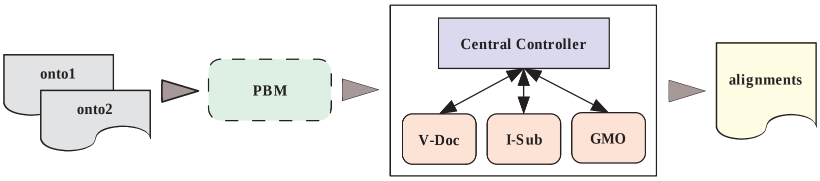


Fig. 1. The architecture of Falcon-AO (version 0.6)

1.2 Specific Techniques Used

To fit the requirements of different application scenarios, we have integrated three distinguishing elementary matchers, V-Doc, I-Sub, and GMO, which are regarded as independent components that make up of the core matcher library of Falcon-AO. Due to the space limitation, we only describe the key features of them. The technical details can be found in the related papers.

- V-Doc [4] discovers alignments by revealing the usage (context) of the domain entities in the ontologies to exploit their intended meanings. More precisely, words from the descriptions of domain entities as well as their neighboring information are simultaneously extracted to form the vectors in the word space, and the similarities between domain entities can be calculated in the Vector Space Model.

- I-Sub [5] is a light-weighted matcher simply based on the string comparison techniques. Its novelty is not only the commonalities between the descriptions of domain entities are calculated but also their differences are examined. Furthermore, it is stable to small diverges from the optimal threshold taking place.

- GMO [1] uses RDF bipartite graphs to represent ontologies, and measures the structural similarities between the graphs by the similarity propagation between domain entities and statements. An interesting characteristic is that GMO can still performs well even without any predefined alignment as input.

More importantly, two major improvements are taken in Falcon-AO (version 0.6). One is the integration of PBM for large-scale ontologies, while the other is the design of central controller.

PBM Due to the size and the monolithic nature of large-scale ontologies, exploiting alignments directly on the whole of them is quite difficult, inefficient, and also unnecessary. We develop an efficient ontology partitioner, PBM [2], to block matching of large-scale ontologies. In PBM, large-scale ontologies are hierarchically partitioned into blocks based on both the structural affinities and linguistic similarities, and then blocks from different ontologies are matched via predefined anchors. The overview of PBM is exhibited in Fig. 2. By applying V-Doc, I-Sub and GMO to the block mappings, we are finally able to generate alignments for large-scale ontologies more quickly while without loss of much accuracy.

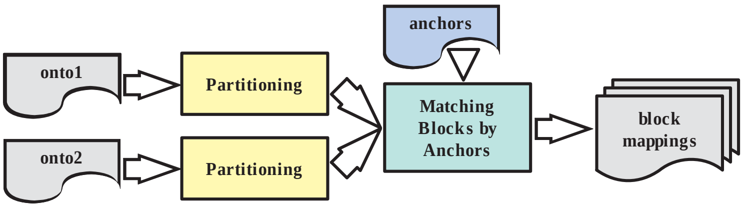


Fig. 2. The overview of PBM

Central Controller As presented above, we have introduced the features of the three elementary matchers, V-Doc, I-Sub and GMO. The question raised naturally here is how to integrate these matchers with ideal performance?

We propose a flexible integration strategy, which depends on the observation of the linguistic comparability as well as the structural comparability. Here, the linguistic comparability is computed by examining the proportion of the candidate alignments against the minimum number of domain entities in the two ontologies.

The calculation of the structural comparability is more complex. It firstly compares the built-in vocabularies used in the two ontologies. The basic assumption is the more built-in vocabularies are mutually included in the two ontologies, the more similar they might be in structure. But only measuring this is inadequate, we also compare the alignments found by V-Doc or I-Sub with high similarities to the alignments discovered by GMO, thus the reliability of the results of GMO can be estimated roughly.

The linguistic and structural comparability can be divided into three categories respectively: low, medium and high. If the comparability is low, it means that the alignments are probably unreliable. If the comparability is medium, the alignments with high similarities would be accepted by Falcon-AO. Otherwise, most of the alignments should be involved into the final output.

When the alignments generated by V-Doc, I-Sub and GMO are obtained, Falcon-AO integrates these alignments by considering the categories of the linguistic and structural comparability, following the rules below:

1. If the linguistic comparability is higher than the structural comparability, the outputted alignments mainly come from V-Doc and I-Sub.
2. If the linguistic comparability is lower than the structural comparability, the outputted alignments largely derived from GMO.
3. Otherwise, the outputted alignments are generated by making a combination among V-Doc, I-Sub and GMO with a weighting scheme.

1.3 Adaptations Made for the Evaluation

We don't make any specific adaptation for the tests in the OAEI 2006 campaign. All the alignments outputted by Falcon-AO are based on the same set of parameters.

1.4 Link to Falcon-AO

The latest version of Falcon-AO (version 0.6) is available at
http://xobjects.seu.edu.cn/project/falcon/matching/resources/falcon.zip,
or http://www.falcons.com.cn/falcon/falcon.zip.

1.5 Link to the Set of Provided Alignments

Full experimental results for all the tests in the OAEI 2006 campaign can be downloaded from http://xobjects.seu.edu.cn/project/falcon/matching/experiments/2006.zip,
or http://www.falcons.com.cn/falcon/2006.zip.

2 Results

The tests provided by the Ontology Alignment Evaluation Initiative (OAEI) 2006 campaign are composed of six categories, including: (a) `benchmark`; (b) `anatomy`; (c) `jobs`; (d) `directory`; (e) `food`; and (f) `conference`. Due to the `jobs` test needs to be further evaluated and discussed, in this section we only present the results of Falcon-AO (version 0.6) in the other five tests, i.e., `benchmark`, `anatomy`, `directory`, `food`, and `conference`.

2.1 `Benchmark`

The `benchmark` test might be divided into five groups: #101–104, #201–210, #221–247, #248–266 and #301–304. The results of Falcon-AO are reported on each group in correspondence. Some more detailed results are listed in Appendix.

#101–104 Falcon-AO performs perfectly on the tests of this group. Please pay attention to #102, Falcon-AO could automatically detect the two candidate ontologies are totally different since both the linguistic comparability and the structural comparability between them are extremely low.

#201–210 Although in this group, some linguistic features of the candidate ontologies are discarded or modified, their structures are quite similar. So GMO takes much effect on this group. For example, in #202, 209, and 210, only a small portion of alignments are found by V-Doc or I-Sub, the rest are all generated by GMO. Since GMO runs much slower, it takes Falcon-AO more time to exploit all the alignments.

#221–247 The structures of the candidate ontologies are altered in these tests. However, Falcon-AO discovers most of the alignments from the linguistic perspective via V-Doc and I-Sub, and both the precision and recall are pretty good.

#248–266 Both the linguistic and structural characteristics of the candidate ontologies are changed heavily, so the tests in this group might be the most difficult ones in all the `benchmark` tests. In some tests, Falcon-AO doesn't perform well, but indeed, in these cases, it is really hard to recognize the correct alignments.

#301–304 Four real-life ontologies of bibliographic references are taken in this group. The linguistic comparability between the two candidate ontologies in each test is high but the structural comparability is moderate. It indicates that the outputs of Falcon-AO mainly come from V-Doc or I-Sub. Alignments from GMO with high similarities are also reliable to be integrated.

The summary of the average performance of Falcon-AO (version 0.6) on the `benchmark` test is depicted in Table 1.

Table 1. The average performance of Falcon-AO on the `benchmark` test

	1xx	2xx	3xx	H-mean	Time
Precision	1.00	0.97	0.98	0.92	472s
Recall	1.00	0.97	0.78	0.86	

2.2 Anatomy

The `anatomy` real world test bed covers the domain of body anatomy and consists of two ontologies, OpenGALEN and FMA, with approximate sizes of several 10,000 classes and several dozens of relations, respectively. By using PBM, Falcon-AO partitions OpenGALEN and FMA into 39 and 407 blocks, separately. Primarily 2,512 alignments are spotted as anchors, and then 42 block mappings are generated. After running further elementary matchers on these block mappings, totally 2,518 alignments are outputted in the end. The complete process takes over 5.5 hours. The experimental results of Falcon-AO (version 0.6) are exhibited in Table 2.

Table 2. The performance of Falcon-AO on the `anatomy` test

	Blocks	Anchors	Pairs	Alignments	Time
OpenGALEN	39	2512	42	2518	5.5h
FMA	407				

Most of these alignments seem credible since the labels of the two entities are the same when they are put into lowercase letters and the punctuation characters are taken out. But due to lack of domain knowledge about the field of anatomy, we couldn't make any further investigation.

2.3 `Directory`

The `directory` case consists of Web sites directories (like Google, Yahoo! or Looksmart). To date, it includes 4,639 matching tasks represented by pairs of OWL ontologies, where classification relations are modeled as *rdfs:subClassOf* relations.

Table 3. The performance of Falcon-AO on the `directory` test

Tasks	Precision	Recall	F-Measure	Time
4369	40.50%	45.47%	42.85%	280s

Falcon-AO is quite efficient in this test, and it only takes less than 5 minutes to complete all the matching tasks. Based on the manual observation, a large portion of generated alignments come from the linguistic perspective, i.e., V-Doc or I-Sub. The precision of Falcon-AO is 40.50%, the recall is 45.47%, and the F-Measure is 42.85%. We also experiment on the previous test set provided by the OAEI 2005 campaign, and the mapping quality seems moderate. The performance of Falcon-AO (version 0.6) on the `directory` test is summarized in Table 3.

2.4 `Food`

The `food` test case includes two SKOS thesauri, AGROVOC and NALT. Since Falcon-AO aims at the Web ontologies expressed in OWL Lite/DL, we firstly transform them into OWL ontologies. The transformation rules are listed as follows. Each concept is transformed into an *owl:Class*. Each broad or narrow relation is transformed into an *rdfs:subClassOf* relation. Each label written in English is reserved. All the other SKOS relations are discarded. Please note that this transformation is incomplete and even sometimes inaccurate.

Table 4. The performance of Falcon-AO on the `food` test

	Blocks	Anchors	Pairs	Alignments	Precision	Time
AGROVOC	1141	11919	253	13009	0.83	5.5h
NALT	950					

Then, Falcon-AO partitions the two corresponding OWL ontologies into 1,141 and 950 blocks, respectively. Supported by 11,919 anchors, Falcon-AO discovers 253 block mappings and runs further elementary matchers on them. Finally, 13,009 alignments are outputted. However, we merely consider the exact matching (equivalence). Currently, the broad or narrow relationship is not addressed in Falcon-AO. The whole process costs nearly 5.5 hours. According to the evaluation by the organizers, the precision is 0.83. The performance of Falcon-AO (version 0.6) is shown in Table 4.

2.5 `Conference`

The collection of tests is dealing with `conference` organization. At present, it includes 45 matching tasks, which are all composed of small ontologies. By comparing to the reference alignments provisionally made by track organizers, the precision of the alignments generated by Falcon-AO is 0.68, while the relative recall is about 0.50. Here, the relative recall is computed as the ratio of the number of all unique correct alignments (sum of all unique correct alignments per one system) to the number of all unique correct alignments found by any of systems (per all systems). In addition, Falcon-AO spends 109 seconds to finish all the matching tasks. Some statistics of the performance of Falcon-AO (version 0.6) are presented in Table 5.

Table 5. The performance of Falcon-AO on the `conference` test

Tasks	Precision	Recall	Time
45	0.68	0.50	109s

3 General Comments

In this section, we summarize some features of Falcon-AO, and discuss the improvement directions towards building a comprehensive ontology alignment system.

3.1 Comments on the Results

Different integration strategies of V-Doc, I-Sub, GMO and PBM lead to significantly different performance of Falcon-AO. In Table 6, we list the most important components that take effect on each test.

Table 6. The most important components integrated in each test

Tests	Components
`Benchmark`	V-Doc, I-Sub, GMO
`Anatomy`	PBM, V-Doc, I-Sub
`Directory`	V-Doc, I-Sub, GMO
`Food`	PBM, V-Doc, I-Sub, GMO
`Conference`	V-Doc, I-Sub

According to the experimental results on these tests shown in the previous section and the integration strategy shown in Table 6, we can analyze some strengths and weaknesses of Falcon-AO (version 0.6) clearly.

Strengths

- Falcon-AO (version 0.6) is a quite flexible ontology alignment tool. It copes with not only ontologies with moderate sizes but also very large-scale ontologies. Moreover, Falcon-AO integrates three distinguishing elementary matchers to manage different alignment applications, and the integration strategy is totally automatic.
- It achieves a good performance in both effectiveness and efficiency. Based on the reference alignments provided by the organizers and the check of human observation, the precision and recall in most cases are sound. Besides, Falcon-AO runs so fast that it only takes a few seconds to complete for ontologies with moderate sizes. Even for large ontologies, it still finishes the alignment tasks in an acceptable time.

Weaknesses

- The tuning of the algorithms within Falcon-AO is still a rigid process. For example, PBM performs well on the large ontologies with simple class hierarchy structures. But when the relations in ontologies are complicated (e.g., OpenGALEN), the partitioning quality of PBM is not sound.
- So far, we do not consider any domain knowledge in the current version of Falcon-AO. Hence, when Falcon-AO meets some applications from specific domains, it might fail to achieve a high quality result.
- Semantic relationship (e.g., equivalence, subsumption) offers general reasoning capability, which is the most prominent difference as compared to schema matching. But currently, Falcon-AO cannot provide alignments with semantic relationship.

3.2 Discussions on the Way to Improve the Proposed System

From the experiments we have learnt some lessons and plan to make improvements in the later versions. The following three improvements should be taken into account.

- While expressing the same thing, people may use synonyms and even different languages. So it is necessary to use lexicons or thesauri in the alignment process.
- The values of parameters used in Falcon-AO is mainly determined by manual setting. Some machine learning approaches can be involved to help automatic adjustment according to different application scenarios.
- The patching strategy for combining the alignments discovered by different matchers needs to be further discussed, e.g., adding some missing alignments, or deleting some wrong and redundant ones.

4 Conclusion

Ontology matching is a crucial task to enable interoperation between Web applications using different but related ontologies. We develop an automatic tool for ontology alignment, named Falcon-AO. From the experimental experience in the OAEI 2006 campaign, we can make a conclusion that Falcon-AO (version 0.6) performs well on most of tests. In our future work, we look forward to making a stable progress towards building a comprehensive ontology alignment system.

References

1. Hu, W., Jian, N., Qu, Y., and Wang, Y.: GMO: A graph matching for ontologies. In Proc. of the K-CAP workshop on Integrating Ontologies. (2005) 41–48
2. Hu, W., Zhao, Y., and Qu, Y.: Partition-based block matching of large class hierarchies. In Proc. of the 1st Asian Semantic Web Conference (ASWC'06). (2006) 72–83
3. Jian, N., Hu, W., Cheng, G., and Qu, Y.: Falcon-AO: Aligning ontologies with Falcon. In Proc. of the K-CAP workshop on Integrating Ontologies. (2005) 85–91
4. Qu, Y., Hu, W., and Cheng, G.: Constructing virtual documents for ontology matching. In Proc. of the 15th International World Wide Web Conference (WWW'06). (2006) 23–31
5. Stoilos, G., Stamou, G., and Kollias, S.: A string metric for ontology alignment. In Proc. of the 4th International Semantic Web Conference (ISWC'05). (2005) 623–637

Appendix: Raw results

Tests are carried out on a PC running Windows XP with an Intel Pentium IV 3.0 GHz processor and 1GB memory.

Matrix of results

In the following table, the results of Falcon-AO in the `benchmark` test are provided with precision (Prec.), recall (Rec.) and machine processing time (Time). Here, the machine processing time is the sum of the time for ontology parsing, ontology matching, alignment generation and evaluation.

#	Name	Prec.	Rec.	Time
101	Reference alignment	1.00	1.00	5.6s
102	Irrelevant ontology	NaN	NaN	3.2s
103	Language generalization	1.00	1.00	2.2s
104	Language restriction	1.00	1.00	2.0s
201	No names	0.96	0.91	2.0s
202	No names, no comments	0.84	0.84	41.2s
203	No comments	1.00	1.00	1.2s
204	Naming conventions	0.96	0.96	1.9s
205	Synonyms	1.00	0.97	2.0s
206	Translation	0.98	0.93	2.0s
207		0.98	0.92	2.2s
208		1.00	1.00	1.1s
209		0.79	0.78	39.6s
210		0.81	0.80	39.2s
221	No specialization	1.00	1.00	1.9s
222	Flattened hierarchy	1.00	1.00	1.9s
223	Expanded hierarchy	1.00	1.00	2.1s
224	No instance	1.00	0.99	1.5s
225	No restrictions	1.00	1.00	1.8s
228	No properties	1.00	1.00	0.9s
230	Flattened classes	0.94	1.00	1.7s
231	Expanded classes	1.00	1.00	2.0s
232		1.00	0.99	1.5s
233		1.00	1.00	0.9s
236		1.00	1.00	0.7s
237		1.00	1.00	1.7s
238		1.00	1.00	2.0s
239		0.97	1.00	0.9s
240		0.97	1.00	1.1s
241		1.00	1.00	0.7s
246		0.97	1.00	0.8s
247		0.97	1.00	1.0s
248		0.86	0.85	38.2s
249		0.85	0.85	37.8s
250		1.00	0.27	0.8s
251		0.55	0.55	43.6s
252		0.71	0.71	42.3s
253		0.86	0.85	36.6s
254		1.00	0.27	0.8s
257		1.00	0.27	0.7s
258		0.56	0.56	43.6s
259		0.72	0.72	42.3s
260		0.90	0.31	0.8s
261		0.80	0.24	0.9s
262		1.00	0.27	0.7s
265		0.90	0.31	0.8s
266		0.80	0.24	0.9s
301	Real: BibTeX/MIT	0.89	0.80	1.5s
302	Real: BibTeX/UMBC	0.90	0.56	0.7s
303	Real: Karlsruhe	0.78	0.73	1.3s
304	Real: INRIA	0.95	0.92	25.4s

Results of the HMatch Ontology Matchmaker in OAEI 2006 *

Silvana Castano[1], Alfio Ferrara[1], and Gianpaolo Messa[1]

DICo Università degli Studi di Milano, via Comelico 39, 20135, Milano, Italy
`{castano,ferrara,messa}@dico.unimi.it`

Abstract. In this paper, we discuss our experience in testing the HMatch matchmaking system by means of the tracks proposed in the ontology alignment evaluation initiative of 2006 [1]. HMatch is a system conceived for the goal of ontology matching in open and distributed systems. It is based on linguistic and structural matching techniques for the evaluation of affinity considering concept names and concept contexts. The paper discusses the results that have been obtained and the possible improvements of the matching techniques in ongoing and future work.

1 Presentation of the system

HMatch is a system for dynamically matching distributed ontologies. It takes two ontologies as input and returns mappings that identify corresponding concepts in the two ontologies, namely the concepts with the same or the closest intended meaning. Mappings are established after an analysis of the similarity of the concepts in the compared ontologies. The similarity analysis is performed through affinity metrics to determine a measure of concept semantic affinity in the range [0, 1]. A threshold-based mechanism is enforced to set the minimum level of semantic affinity required to consider two concepts as matching concepts. HMatch is part of the Helios framework [4], conceived for supporting knowledge sharing and ontology-addressable content retrieval in peer-based systems.

A more detailed description of HMatch can be found in [3].

1.1 State, purpose, general statement

With respect to the different purposes of matching, the state of HMatch is the following:

- *Ontology matching* is the original purpose of HMatch which has been designed with the goal of working with the different languages of OWL (i.e., OWL Lite, OWL DL, and OWL Full) [11].

---

* This paper has been partially funded by BOEMIE, FP6-027538 - 6th EU Framework Programme and by ESTEEM MIUR PRIN project funded by the Italian Ministry of Education, University, and Research.

[1] http://oaei.ontologymatching.org/2006/

- *Schema matching*. In developing HMatch, we started from the schema matching functionalities of Artemis integration system [2]. From Artemis we borrowed the thesaurus-based approach for name affinity management, but we made a number of extensions for matching linguistic features of ontology elements in order to rely only on the WordNet lexical system, to provide a fully-automated matching process. Furthermore, we have moved from the notion of structural affinity, typical of schema elements based on attributes, to the notion of *contextual affinity*, typical of ontology elements, based on semantic relations with explicit semantics, with consequent development of suitable contextual affinity evaluation techniques.
- *Version matching*. Currently, we are extending HMatch towards version matching in the context of the BOEMIE European Project [1] Specifically, we are extending the tool to perform instance matching and to evaluate the differences between different versions of the same ontology to support the evolution of multimedia ontologies.
- *Directory matching*. HMatch can perform directory matching in the deep matching model configuration, by considering taxonomic knowledge in the directory as is-a relations in all cases. However, directory taxonomic relations have a different semantics (e.g., *part-of*, *contain*), and a manual pre-processing is required in order to distinguish them in the matching process.

1.2 Specific techniques used

Given two concepts, HMatch calculates their semantic affinity value as the linear combination of a *linguistic affinity* value and a *contextual affinity* value. The basic techniques used in HMatch are linguistic and structure-based techniques that are applied to concept names and contexts. For a more detailed classification of these and other techniques the reader can refer to [6].

Linguistic-based affinity techniques. Linguistic techniques consider names of ontology elements and their meaning. To capture the meaning of names for ontology matching, a thesaurus of terms and weighted terminological relationships is exploited. In HMatch, the thesaurus is automatically derived from the lexical system WordNet [8]. The thesaurus is structured as a graph, where the nodes represent terms and the edges represent terminological relationships. Terminological relationships represented in the thesaurus are SYN, BT, NT, and RT. SYN (synonymy) denotes that two terms have the same meaning. BT (broader term) (resp., NT (narrower term)) denotes that a term has a more (resp., less) general meaning than another term. Finally, RT (related terms) denotes that two terms have a generic positive relationship. A weight W_{tr} is associated with each terminological relationship $tr \in \{\mathsf{SYN}, \mathsf{BT/NT}, \mathsf{RT}\}$ in the thesaurus. Such a weight expresses the implication of the terminological relationship for semantic affinity. Different types of relationships have different implications for semantic affinity, with $W_{\mathsf{SYN}} \geq W_{\mathsf{BT/NT}} \geq W_{\mathsf{RT}}$. Given the thesaurus of weighted terminological relationships, the linguistic affinity is evaluated by means of a term affinity function $\mathcal{A}(t,t') \rightarrow [0,1]$ which evaluates the affinity between two terms t and t'. $\mathcal{A}(t,t')$ of two terms t and t' is equal to the value of the highest-strength path of terminological relationships between them in Th if at least one path exists, and is zero otherwise. A path strength is computed by multiplying the weights associated with each terminological relationship involved in the path, that is:

$$\mathcal{A}(t, t') = \begin{cases} \max_{i=1\ldots k} \{W_{t \to_i^n t'}\} & \text{if } k \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where: k is the number of paths between t and t' in Th; $t \to_i^n t'$ denotes the ith path of length $n \geq 1$; $W_{t \to_i^n t'} = W_{1_{tr}} \cdot W_{2_{tr}} \cdot \ldots \cdot W_{n_{tr}}$ is the weight associated with the ith path, and $W_{j_{tr}}, j = 1, 2, \ldots, n$ denotes the weight associated with the jth terminological relationship in the path.

Structure-based affinity techniques. Structure-based techniques consider properties and concepts directly related to a concept c through a semantic relation in an ontology. Given a concept c, we denote by $P(c)$ the set of properties of c, and by $C(c)$ the set of concepts that participate in a semantic relation with c (namely, its *adjacents*). The context of a concept in HMatch is defined as the union of the properties and of the adjacents of c, that is, $Ctx(c) = P(c) \cup C(c)$. Also contextual features are weighted in HMatch. In particular, we associate a weight W_{sp} to strong properties, and a weight W_{wp} to weak properties, with $W_{\mathsf{sp}} \geq W_{\mathsf{wp}}$ to capture the different importance they have in describing the concept. In fact, strong properties are mandatory properties related to a concept and they are considered more relevant in contributing to concept description. Weak properties are optional for the concept in describing its structure, and, as such, are given less importance. Each semantic relation has associated a weight W_{sr} which expresses the strength of the connection expressed by the relation on the involved concepts. Considering the semantic relations of OWL, we have the weights $W_{\mathsf{equivalence}} \geq W_{\mathsf{subClassOf}}$. The greater the weight associated with a semantic relation, the higher the strength of the semantic connection between concepts. Given two elements e and e' in the contexts of c and c', respectively, their affinity is calculated according to the following function $\mathcal{C}(e, e') \to [0, 1]$:

$$\mathcal{C}(e, e') = \mathcal{A}(n_e, n_{e'}) \cdot (1- \mid W_e, W_{e'} \mid \tag{2}$$

where n_e and $n_{e'}$ denote the names of e and e', respectively, while W_e and $W_{e'}$ denotes the weights associated with e and e'. As an example, suppose that we compare two concept contexts $Ctx(c)$ and $Ctx(c')$ both containing the property $author$ that is a strong property (i.e., featured by a minimum cardinality greater than or equal to 1) in the first context and a weak property (i.e., an optional property) in the second context. Thus, by using a weight equal to 1.0 for strong properties and equal to 0.5 for weak properties, we obtain:

$$\mathcal{C}(author_{Ctx(c)}, author_{Ctx(c')}) = \mathcal{A}(author, author) \cdot (1- \mid 1.0 - 0.5 \mid) = 0.5$$

since $\mathcal{A}(author, author) = 1.0$.

Given two concepts c and c', the comprehensive semantic affinity $SA(c, c')$ is calculated as the weighted sum between their linguistic affinity $LA(c, c')$ and their contextual affinity $CA(c, c')$, as follows:

$$SA(c, c') = W_{la} \cdot LA(c, c') + (1 - W_{la}) \cdot CA(c, c') \tag{3}$$

where $W_{la} \in [0, 1]$ weights the relevance of the linguistic affinity in matching evaluation. The two measures of linguistic affinity $LA(c, c')$ and $CA(c, c')$ are calculated in a different way depending on the matching model that is selected in the configuration of HMatch.

Matching models. Four matching models have been conceived to span from surface to intensive matching, with the goal of providing a wide spectrum of metrics suited for dealing with many different matching scenarios that can be encountered in comparing real ontologies, such as OWL ontologies. The main difference among the four matching models is the composition of the context. In the *surface model*, the context is not considered limiting to linguistic affinity. In the *shallow model*, only properties and property restrictions are considered for concept context. In the *deep model*, we consider both properties and semantic relations, such as taxonomic relations. Finally, in the *intensive model* we consider the whole context, by taking into account also the property ranges and values. For all the models the linguistic affinity $LA(c, c')$ between two concepts c and c' is calculated to be equal to the function $\mathcal{A}(n_c, n_{c'})$, where n_c and $n_{c'}$ denote the names of c and c', respectively. For the contextual affinity evaluation, we provide two main strategies, namely the *standard* strategy and the *Dice coefficient* strategy. The standard strategy produces a non-symmetric contextual affinity measure. For each element e in the source concept context $Ctx(c)$, we search for the best matching element e' in the target concept context $Ctx(c')$ by exploiting the function $\mathcal{C}(e, e')$ described above. Given the best matching value m_e found for e with respect to the elements in the context of c', the comprehensive contextual affinity is calculated as follows:

$$CA(c, c') = \frac{\sum_{e_i \in Ctx(c)} m_{e_i}}{\mid Ctx(c) \mid}$$

where $\mid Ctx(c) \mid$ denotes the number of elements in c.

According to the Dice coefficient strategy, the contextual affinity is calculated as follows:

$$CA(c, c') = \frac{\mid x \in Ctx(c) \cap Ctx(c') \mid}{\mid x \in Ctx(c) \cup Ctx(c') \mid}$$

where $\mid x \in Ctx(c) \cap Ctx(c') \mid$ denotes the number of matching elements in $Ctx(c)$ and in $Ctx(c')$, that is the number of elements having a value of $\mathcal{C}(e, e')$ higher than a given matching threshold.

1.3 Matching policies

Since HMatch has been developed with the goal of achieving a high level of flexibility and configurability of the matching process, a *matching policy* P has be set, which is defined as follows:

$$P = \langle W_{la}, T, M, C, I, S, E \rangle$$

where: W_{la} is the weight associated with the linguistic affinity; $T \in [0, 1]$ denotes the threshold used for selecting matching results; $M \in$ {surface, shallow, deep, intensive} denotes the matching model; $C \in$ {one-to-one, one-to-many} denotes the matching cardinality; $I \in$ {true, false} denotes if the context elements inherited through the taxonomic relations are to be considered in the matching process; $S \in$ {standard_strategy, dice_strategy} denotes the metrics used for the contextual affinity evaluation; $E \in$ {empty_pessimistic, empty_neutral, empty_optimistic} denotes the strategy to be enforced to handle empty contexts. Using the pessimistic strategy, the contextual affinity value is set to 0, to mean that no matching elements have been found in their contexts. In the neutral strategy the empty contexts are considered to have a semantics analogous to the one of the NULL value in relational databases; the contextual affinity is set to undefined to capture this semantics. In the optimistic strategy, the contextual affinity value is set to 1, to mean that two empty contexts are considered to fully match.

1.4 Adaptations made for the evaluation

For the purposes of the OAEI 2006 initiative, we adopted the standard implementation of HMatch as a Protégé[2] plugin. This version adopts the Protégé OWL API[3] and is fully integrated into the Protégé framework. We only introduce a command line version in order to use HMatch as an independent tool, especially for the benchmark. We have implemented two main extensions specifically conceived for the contest. The first extension is the support for the output Alignment format required by the organizers in addition to the native HMatch results format. The second extension regards the evaluation of the linguistic affinity. We introduced a new facility of HMatch that performs linguistic affinity evaluation using a n-gram algorithm [5]. This technique, being syntactic is faster than the thesaurus-based analysis, thus overcomes some scalability problems that we noticed with very large ontologies, such as in the case of *anatomy* and *directory-full*.

1.5 Link to the system and parameters file

The HMatch implementation used for the contest together with the policy used for configuration and the results can be find at:
http://islab.dico.unimi.it/OAEI2006/islab_results.html.

2 Results

All the results have been obtained by configuring HMatch with the following policy:

Policy		Value
W_{la}	Linguistic affinity weight	0.5
T	Threshold	0.6
M	Matching model	Deep
C	Matching type	One-to-One
I	Inheritance	True
E	Empty context strategy	Neutral
S	Contextual affinity strategy	Dice coefficient

---

[2] http://protege.stanford.edu/

[3] http://protege.stanford.edu/plugins/owl/api/

In particular, the most relevant parameters are i) the matching model, since the deep model forces HMatch to consider both properties and semantic relations in the concept contexts, ii) the weight for linguistic affinity, since the value 0.5 determines a perfect balance between the linguistic affinity evaluation and the contextual affinity evaluation, and iii) the threshold, which is used for cutting off the results that are not considered relevant in the matching case. We have tested several possible configurations of HMatch on the benchmark. In some matching cases there are other configuration policies that produce better results in terms of precision and recall than the one we have chosen. The actual choice was motivated by the fact that we considered the various tasks proposed in the contest with the goal of configuring HMatch with a policy that could guarantee a generally satisfactory behavior of the system in the different matching cases. In particular, we have tested HMatch on all the test cases provided in the contest, with the goal of receiving a feedback about the application of the system to different and highly heterogeneous matching cases.

2.1 Benchmark

Obtained results on the proposed benchmark are strongly affected by the fundamental role that the ontology linguistic features play in the HMatch matching process. In fact, we obtained an average precision value of 0.84 and an average recall value of 0.55. These results are influenced by the fact that we obtained poor result for the ontology cases where the concept and property labels were substituted with randomly generated strings of characters. The difference between precision and recall values when we consider all the cases is due to the fact that, in some of the randomly generated ontologies (e.g., case 259), there is a property which maintains the original name (i.e., lastName). This matching is retrieved by HMatch and it increases the precision of the results. Another issue that affects the results quality, from the linguistic point of view, is the presence of matching cases where the concept and property labels are french terms. In these cases, since some of the properties match, we obtained precision values about 0.4 and recall values about 0.2. The benchmark results are also useful to suggest possible improvements of HMatch, with the goal of addressing also the anomalous cases where the linguistic information is completely missing due to the design choices.

2.2 Anatomy

With the anatomy track, obtained results suggest the following considerations. Due to the domain specific terminology used in the ontologies, either using the WordNet thesaurus or a string matching technique, the results are affected by the fact that the concepts are labeled with long strings describing specific terms. In the case of a domain specific terminology, the linguistic matching would benefit from the availability of specific thesauri. Given the large amount of data in the two compared ontologies, the string-matching procedure for linguistic affinity is more suitable, while affecting the capability of the system to capture the semantics of the terms used in the two ontologies that would instead be possible using the thesaurus. Moreover, the openGalen ontology has a anomalous OWL structure, since OWL classes are used as meta-classes, while individuals represent the domain concepts. For this reason we needed a wrapper to compare the FMA concepts with the concepts of openGalen, and only the linguistic comparison was possible.

2.3 Directory

The directory matching is a new task for HMatch, which was not originally designed for dealing with peculiar features of directory repositories. In particular, two main characteristics of directory taxonomies require specific support not directly provided by HMatch. The terminology used for labeling the directories is often affected by the structure of the taxonomy itself more than by the subject of the directory. Examples of this terminology is given by terms like A-H that is referred to the alphabetic order more than to the subject of the directory, or African_2 where the name of the directory is associated with information about the number of equivalent directories in the taxonomy. A second problem is given by the taxonomy itself. In fact, HMatch gives the is-a semantics to the OWL sub-class relations as in formal ontologies. Although, the sub-directory relations represented as OWL sub-class relations have in fact different meanings. For example, we have a sub-class relation between Animal_Webcams and Space_and_Science that denotes a generic positive relation between the two concepts rather than an is-a relation. Another example is given by the sub-class relation between California and United_States that denotes a geographical part-of relation.

2.4 Food

The food track requires to match two XML thesauri. We developed a wrapper from the SKOS XML format to OWL in order to match the thesauri with HMatch. The track requires also to recognize different kind of mapping relations between the source and the target, i.e., exactMatch, broadMatch, and narrowMatch. Using HMatch, we provide a measure of the semantic affinity between two concept, that is a measure of the fact the the two terms have the same meaning. Because of this reason, we provided only an evaluation of the exactMatch mapping between the two ontologies. In order to evaluate the broad and narrow matching relations, the thesaurus component of HMatch could be exploited, but this has not been done due to the contest requirement of using the same techniques for all different cases.

3 General comments

One of the main issues in the field of ontology matching is the need of flexible algorithms and tools, capable to adapt to different domains and also to different interpretation of the notions of *alignment* and *similarity*. Some of these differences depends on the concept descriptions provided by the ontologies to be compared with their specific level of semantic complexity. The choice of the best approach or the best combination of approaches depends on the specific matching case we are dealing with and on the domain of the ontologies. For example, formal ontologies can benefit from a logic approach, while thesauri and dictionaries require a deep linguistic analysis; finally, structure affinity is suitable for directories and repositories. The domain affects also the kind of techniques that are used as well as the matchmaking utilities (e.g., thesauri, external sources, type of mapping relations) that are involved in the matching process. A good example is given by the *anatomy* track of the contest. In this domain, we work with a specific and domain dependent terminology that requires a specific linguistic analysis. A second example is given by the matching of directories or also by the matching of spatial or temporal ontologies. In this cases, in fact, some properties or relations should be matched by using specific matching operators. For example, the property *author* and the property *below* have a different role on concept definition when used in a spatial domain, even if they are represented by means of the same language construct. The matching should take into account all these specific requirements by adapting the matching process and the matching operations to the specific domain that is taken into account.

3.1 Comments on the results

The results obtained in the OAEI tasks show how HMatch can provide a good balance in the results between precision and recall with a fully automated matching that does not require any specific external source neither in terms of a training set of results nor in terms of domain specific thesauri. Although, if on a side this characteristic means that HMatch can be used in several different scenarios, on the other side, it shows a limitation of the system in working either with very specific domain ontologies or with ontologies in which the linguistic information is missing. Some other limitations regard the scalability of the linguistic techniques adopted by the system is the case of large ontologies. To overcome this limitation, we have implemented for the purpose of the contest a new string matching functionality. The main considerations that we can make based on matching cases and obtained results experienced are the following.

1. *Linguistic features*: the terminology used for naming and labeling concepts and properties is an important aspect of ontologies and provides information to conclude the similarity between the ontology elements. We are conscious that, In many cases, it is not sufficient alone, also because they embed a subjectivity element, deriving from who has been designed the ontology. However, the linguistic features are undoubtedly an important starting point also for deriving a first set of mappings to be refined by exploiting other kinds of matchings.
2. *Structural features*: concepts can be similar also in terms of their structure. The structure is seen in terms of the links that connect different concepts and also as the number and type of properties that characterize each concept. It is important to note that the structure evaluation does not refer to the semantics of the concept relations and properties. For example, in the directory taxonomies the semantics of the *sub-category* relation is not ever well defined and can denote many different real relations among categories, e.g., containment, is-a, part-of. In this case, the structure of the taxonomies that are considered is the key feature for detecting the similarity of the concepts, more than the relation semantics.
3. *Logical features*: from the logical point of view, the ontology matching should consider the formal semantics of the ontologies to be compared in order to i) evaluate the consistency between the mappings and the concept descriptions, ii) apply deductive reasoning to retrieve new mappings starting from an initial set of mappings

(e.g., manually provided or retrieved by means of other techniques), iii) provide an interpretation of the resulting mappings.

3.2 Discussions on the way to improve the proposed system

By analyzing the results obtained in the different tracks, together with the general comments discussed in the previous section, a first improvement that can be introduced in HMatch is to emphasize the distinction among the linguistic, structural, and logic approaches to ontology matching. HMatch is based mainly on linguistic features. We believe that linguistic matching is a fundamental component for a semantic matchmaker, but we noticed that, in some cases, structure and logics of the ontologies to be compared should be considered with no reference to the ontology element names. Another important direction for improving HMatch is to emphasize the need of different metrics in order to take into account the specific features of the different ontology domains. HMatch provides four different matching models to address the fact that different ontologies can vary with respect to their semantic complexity and with respect to their structure. A further improvement in this direction is to support specific relations in the matching process, such as spatial or temporal relations.

3.3 Comments on the OAEI 2006 procedure

The OAEI 2006 procedure is well suited to give to matching researchers a complete feedback about their work. Although, we believe that the requirement of using only one set of parameters for the whole contest was a strong limitation, especially because some of the test cases (i.e., anatomy, food) have peculiar features that would benefit from a more flexible configuration. We believe that the capability of matching algorithms to be configured in order to deal with different scenarios is a key feature for ontology matching, but the flexibility cannot be appreciated using the same configuration. If the goal is to test generic-purpose algorithms, the test cases should be more homogeneous with respect to the ontology type and domain. Otherwise, it should be possible to modify the algorithms configuration for the different cases.

3.4 Comments on the OAEI 2006 test cases

The only comment we have is that, at the end of the evaluation phase, would be useful to have the expected results also for the blind tests, in order to improve the algorithms used where required.

3.5 Comments on the OAEI 2006 measures

The traditional precision and recall measures seem to be the most suitable for the matching result evaluation. Although, these measures should be calculated in a flexible way. For example, we should allow the algorithms to provide mappings also among external elements that are imported in the ontologies.

3.6 Proposed new measures

A simple suggestion for new measures is referred to the need of taking into account the time of computation in the matching evaluation. The idea is to combine the computation time with precision and recall, in order to measure the trade-off between time performances of the algorithms and quality of the results.

4 Conclusion

The experience of the OAEI 2006 contest was extremely useful as a feedback about the design and implementation of the current version of HMatch. We had some confirmation of the results obtained in the previous tests, but we had also some new helpful tip about possible improvements of the approach and related techniques. In particular, our future work will be devoted to: i) study new matching techniques that could be used in combination with the linguistic techniques of HMatch, in order to improve the flexibility of the system with respect to different matching scenarios; ii) address new purposes of the matching, such as directory of ontology version matching, by studying specific metrics and techniques for these cases; iii) implement and test a new version of HMatch in the context of the BOEMIE project, where our matchmaking system is used for the purpose of ontology evolution.

References

1. BOEMIE Project. Bootstrapping ontology evolution with multimedia information extraction. IST-2004-2.4.7 Semantic-based Knowledge and Content Systems Proposal, 2004. http://www.boemie.org.
2. S. Castano, V. De Antonellis, and S. De Capitani Di Vimercati. Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering*, 13(2):277–297, March/April 2001.
3. S. Castano, A. Ferrara, and S. Montanelli. Matching ontologies in open networked systems: Techniques and applications. *Journal on Data Semantics (JoDS)*, V, 2006.
4. S. Castano, A. Ferrara, and S. Montanelli. *Web Semantics and Ontology*, chapter Dynamic Knowledge Discovery in Open, Distributed and Multi-Ontology Systems: Techniques and Applications. Idea Group Publishing, Hershey, Pennsylvania, USA, 2006.
5. Y. S. Ching. n-gram statistics for natural language understanding and text processing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1:164–172, 1979.
6. Alfio Ferrara. *Matching of Independent Ontologies in Open Networked Systems: Methods and Techniques*. PhD thesis, Università degli Studi di Milano, 2005.
7. Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review Journal*, 18(1), 2003.
8. G. A. Miller. WordNet: A lexical database for english. *Communications of the ACM (CACM)*, 38(11):39–41, 1995.
9. N. Noy. Semantic integration: a survey of ontology-based approaches. *SIGMOD Record Special Issue on Semantic Integration*, December 2004.
10. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics (JoDS)*, 1, 2005.
11. M. K. Smith, C. Welty, D. L. McGuinness, and (eds.). Owl web ontology language guide, 2004. World Wide Web Consortium (W3C), http://www.w3.org/TR/owl-guide/.

JHU/APL Onto-Mapology Results for OAEI 2006

Wayne L. Bethea, Clayton R. Fink, John S. Beecher-Deighan

Research and Technology Development Center
Johns Hopkins University Applied Physics Laboratory
Laurel, MD 20723, USA
Wayne.Bethea@jhuapl.edu

Abstract. Numerous techniques for ontology alignment and mapping have appeared in the literature, but there has been little discussion on the use of formal semantics for the task. Typical solutions apply multiple techniques to produce their results. We demonstrate that a hybrid solution that brings together a number of matching techniques yields the best results. An essential component of any ontology mapping solution is the ability for users to interact with the system and manipulate intermediate and final results. We introduce Onto-Mapology; an approach to ontology mapping that integrates techniques based on string/text matching, structure/graph matching, and semantic (rule-based/logic-based) matching. After the initial design, development, testing and evaluation we applied Onto-Mapology to the OAEI 2006 test cases with encouraging results.

1 Onto-Mapology: The Mapping Process

Ontology mapping techniques have been discussed in the literature that describe string and text matching techniques [1], schema matching techniques [2], categorical information mapping techniques [3], and machine learning techniques [4], but very little has been discussed that describes formal semantic matching techniques. Onto-Mapology is the Johns Hopkins University Applied Physics Lab (JHU/APL) ontology mapping software solution that was designed and developed with strong consideration for human participation in the mapping process. It integrates techniques based on string/text matching, structure/graph matching, and semantic (rule-based/logic-based) matching. It allows users to apply different combinations of these techniques, or a hybrid algorithm that produces solid results in our testing. This paper discusses Onto-Mapology, our approach to the ontology mapping process, and our results for OAEI 2006.

1.1 Purpose, General Statement

We determined at an early stage in the design of our mapping solution that given the state of the art in ontology mapping, human participation would be a crucial part of any successful real world application. This meant that we needed to design user interaction as an important part of the software design rather than an afterthought.

Onto-Mapology was developed as an Eclipse Plug-in, where Eclipse.org is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software [5]. The Eclipse SDK is a development environment that many software developers and end users are familiar with, and provided us with the user interface and the environment to offer important user interactions.

Upon reviewing the literature it was clear that there was not much discussion on using formal semantics (e.g. using reasoning engines and inference) in the ontology mapping process. It is appropriate to hypothesize that this is due to the fact that much of the semantic meaning expressed in past and present ontologies is expressed through the linguistic content. In contrast, as ontologies mature and users get better at using the tools at their disposal for creating and maintaining ontologies, we will begin to see more semantically expressive ontologies. We wanted to determine what would be needed to successfully utilize formal semantics to help accomplish ontology mapping, and we wanted to implement some semantic matching techniques in Onto-Mapology. We have identified the features of formal semantics expressed in ontologies that will improve the results of ontology mapping dramatically using future ontologies.

In current ontologies, the majority of the information available for communicating semantic meaning, and thus for matching and mapping, is in the text of the ontologies. Many ontology mapping solutions rely predominantly on matching techniques performed on the textual content of ontologies, and that is where we started identifying and implementing our matching techniques.

As ontologies become more structurally sophisticated, or as textual content becomes more degraded, structure matching techniques can play increasingly significant roles in ontology matching. Also, in the literature there is a long tradition of supplementing text matching techniques with structure or graph matching techniques, and several approaches are described in the references provided above and the following [6, 7, 8]. In addition to envisioning ontologies becoming more structurally sophisticated one can envision ontologies becoming more semantically expressive. At present the majority of the ontologies that have been developed or are available in the public domain are not very rich semantically. They rely largely on capturing and conveying meaning through linguistic content. But the vision of the Semantic Web implies that the ontologies will be sufficiently expressive as to allow software agents on the Web to act “intelligently” [9].

Here we have provided the motivation and goals for the design and development of Onto-Mapology; we only had to make some minor adjustments to apply the OAEI 2006 benchmark test cases.

1.2 Specific Techniques Used

The mapping solution integrates techniques based on string matching, structure matching, and semantic matching. As we discuss our matching techniques we are assuming the ontologies are expressed using OWL.

1.2.1 Linguistic Matching Techniques

Onto-Mapology can use an implementation of any string comparison matching algorithm, as long as the implementation can use a provided abstract interface. We implemented the algorithms from the SecondString [10] project by creating wrappers around the SecondString string comparison classes. These classes include the Jaro, Jaro-Winkler, TFIDF, and Monge-Elkan string similarity algorithms. Our testing yielded the Jaro-Winkler algorithm as the best performer of the SecondString classes in our implementation. This algorithm calculates the edit distance between two strings and captures the string similarity using:

$$Jaro(s,t) = \frac{1}{3} \cdot \left(\frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - T_{s',t'}}{2|s'|} \right)$$

$$Jaro\text{-}Winkler(s,t) = Jaro(s,t) + \frac{P'}{10} \cdot (1 - Jaro(s,t))$$

For two strings *s* and *t*, let *s'* be the characters in *s* that are “common with” *t*, and let *t'* be the characters in *t* that are "common with" *s* … Let $T_{s',t'}$ measure the number of transpositions of characters in *s'* relative to *t'*. *P* is the length of the longest common prefix of *s* and *t*, and *P' = max* (*P*, 4) [11].

Onto-Mapology also implements an algorithm that uses an *n=2* n-gram comparison, affectionately known as “How to Strike a Match” [12]. It bases the similarity score on a comparison of consecutive letter pairs in the two strings. This approach bases its metric upon the similarity of adjacent letters within a string. It meets the following two criteria; 1) strings with slight discrepancies will be scored as similar; 2) strings that contain the same words but differ in arrangement will be scored as similar. This algorithm captures the string similarity using:

$$similarity(s1,s2) = \frac{2 \times |pairs(s1) \cap pairs(s2)|}{|pairs(s1)| + |pairs(s2)|}$$

For two strings *s1* and *s2* the similarity is twice the number of character pairs that are common to both strings divided by the sum of the number of character pairs in the two strings. When using a single matching technique in Onto-Mapology this algorithm tended to yield the best results on the OAEI 2006 test suite.

Onto-Mapology also uses the Lucene [13] text search engine and indexing tool to create a matcher that compares the terms in two ontologies based on the content of their comments, labels and local names. Lucene is high-performance, scalable, full-featured, open-source, and written in Java. We index one ontology using Lucene, treating each term as a "document" and the term's local name, comment text and label text as the document's content. Lucene removes all stop words from the text and creates an index organized by term. Subsequent ontologies are processed term by term, and each term's local name, comment text and label text are processed using Lucene's string processing capabilities to remove all stop words. The resulting list of words is then used as a search argument against the index created from the first ontology. Lucene is configured to use a letter distance algorithm to score the hits against the index. We treat a high scoring hit as a match between a term in one ontology with a term in another ontology.

1.2.2 Structure Matching Techniques

Onto-Mapology implements an algorithm called "Neighborhood Match" where each ontology is viewed as a graph with nodes and edges, the nodes are classes (or data types) and the edges are properties. For each node in the respective graphs the similarity between nodes (ontology terms) is determined by the number of nodes and edges from each nodes "neighborhood" that match. The neighborhood is determined by specifying how many edges the algorithm should traverse from the starting node. The match is determined by the type of the node or edge or by the text of the node or edge if the user wants to use an algorithm that combines text matching techniques and structure matching techniques.

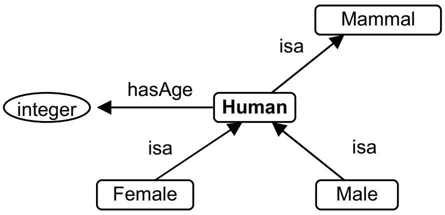


Figure 1: Human Node Neighborhood

So, from a starting node in each of the graphs, the algorithm follows all edges leading from those nodes and compares the edges and related nodes. For example, using the **Human** node in figure 1 the neighborhood 1 edge away would be the *subClass* property relating to **Mammal**, the *subClass* property relating to **Male**, the *subClass* property relating to **Female**, the *hasAge* property relating to **integer**, the classes **Mammal**, **Male**, **Female**, and the data type **integer**. Using the **HomoSapien** node in figure 2 the neighborhood 1 edge away would be the *subClass* property relating to

Mammalian, the *subClass* property relating to **Female**, the *subClass* property relating to **Male**, the *hasAge* property relating to **integer**, the classes **Mammalian**, **Female**, **Male**, and the data type **integer**.

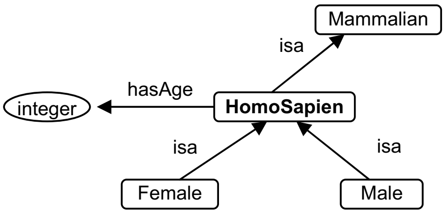


Figure 2: HomoSapien Node Neighborhood

In a purely structural context, our algorithm would compare the 3 subClass properties, 1 data type property, 3 classes, and 1 integer data type neighborhood of the **Human** node to the 3 subClass properties, 1 data type property, 3 classes, and 1 integer data type neighborhood of the **HomoSapien** node and find a match. In a combined linguistic and structural context, our algorithm would also compare the strings of the neighborhoods. For example, it would compare "Mammal," "Male," "Female," and "hasAge" from the Human node with "Mammalian," "Female," "Male," and "hasAge" from the HomoSapien node. In this example, text matching techniques would not produce a match between Human and HomoSapien where structure matching would.

1.2.3 Semantic Matching Techniques

Jena includes a general purpose rule-based reasoner which is used to implement both the RDFS and OWL reasoners but is also available for general use. This reasoner supports rule-based inference over RDF graphs and provides forward chaining, backward chaining and a hybrid execution model. A rule for the rule-based reasoner is defined by a Java Rule object with a list of body terms (premises), a list of head terms (conclusions) and an optional name and optional direction. Each term or **ClauseEntry** is either a triple pattern, an extended triple pattern or a call to a built-in primitive. A rule set is simply a List of Rules.

Onto-Mapology implements rules based on class hierarchy and property hierarchy. For example, we have a rule that states if a class in one ontology is determined to be equivalent to a class in another ontology then the super classes of the equivalent classes are equivalent. The rule looks like this:

```
(?a owl:equivalentClass ?b), notEqual(?a, ?b),
(?a rdfs:subClassOf ?c), (?b rdfs:subClassOf ?d),
notEqual(?c, ?d), notBNode(?c), notBNode(?d) ->
```

(?c owl:equivalentClass ?d)

We also have a rule that states that if the domain and range of a property in one ontology are determined to be equivalent to the domain and range of a property in another ontology, respectively then the properties are equivalent. The rule looks like this:

```
(?a rdfs:domain ?b), (?c rdfs:domain ?d),
(?b owl:equivalentClass ?d),
(?a rdfs:range ?e), (?c rdfs:range ?f),
(?e owl:equivalentClass ?f) ->
(?a owl:equivalentProperty ?c)
```

Semantic matching through rules doesn't fully access the formal semantics expressed in the ontologies. For sufficiently expressive ontologies an OWL DL reasoning engine should be able to indicate terms that are equivalent and terms that are not equivalent because of the expressed formal semantics. In order for Onto-Mapology to exploit formal semantics expressed in ontologies to assist in ontology alignment we have incorporated Pellet [14], an open-source Java based OWL DL reasoner, into our solution.

1.2.4 Hybrid Algorithm

The Onto-Mapology hybrid algorithm first generates a list of alignments based on name equivalence (100% similarity) using the Jaro-Winkler matching technique. Terms matched in this way are placed into a "high confidence" list. Terms in this list can not be matched again. Next, alignments are created using the lemma matching technique and matched terms are added to the high confidence list. Alignments made in this step do not consist of matches created during the Jaro-Winkler phase. Finally the remaining terms in each ontology are compared based on type. If two terms are the same type then they are compared both structurally and semantically.

Structural comparison is performed as follows: if two terms share 80% equivalent neighborhoods they are judged to be equivalent. Two neighbors are judged to be equivalent if they have been aligned previously or if they share the same type. Semantic equivalence is based upon OWL language relations. We define properties to be equivalent if they have had their domains and ranges aligned. For classes, we state that if two classes share equivalent child class then they are defined to be equivalent. We have completed the task of bringing these techniques together in one algorithm, but we need to add the formal semantic reasoning and characterize which parts of the algorithm will work best under which circumstances. After we have the full implementation and the characterization we can fine tune the algorithm to give the best results given multiple and different types of ontologies.

2 OAEI 2006 Results

Here we present the results of alignment experiments performed on the OAEI 2006 campaign. All the output is produce using the same input parameters. In the presentation of our results and analysis of our algorithms we have also included our experiment results from the OAEI 2005 benchmark tests. The OAEI 2005 based experiment results used linguistic matching techniques to establish alignments based on name similarity. These results were not submitted to the OAEI 2005 campaign because we had not known about the OAEI until after the submission deadline. We will not discuss the OAEI 2005 results or algorithms any further in this paper.

2.1 Benchmark

The benchmark test cases are broken up into five main categories. The first series of tests (#101-104) examine an algorithm's ability to make basic matches. It also determines the program's ability to handle discrepancies of OWL Language usage, like generalization and restriction.

In this first grouping of tests we found our algorithm to be relatively successful in obtaining satisfactory results. However, we found that test #102 created problems for our algorithm. In this test case we compare the reference ontology to one that is irrelevant. The string similarities of the terms in each document are quite different; this leads our structure matching component to become more prevalent thus causing a precision of 0 to occur when any mappings were made. The average performance of this group is depicted below:

	Precision	Recall	F-Measure
Average 2005	0.81	0.99	0.89
Average 2006	0.75	1.00	0.75

The next series of tests (#201-266) manipulate six parameters: name, comments, specialization hierarchy, instances, properties, and classes. These tests allow for algorithms to be examined in specific situations. This set of tests was the most useful to us; they allowed us to see the specific areas where we need improvement.

Tests (#201-210) manipulate names and comments. In this set of test cases our algorithm performed relatively well except in those cases where name similarity was not high (#201, 202, 209, & 210). Even in those cases our recall was still quite high.

	Precision	Recall	F-Measure
Average 2005	0.64	0.28	0.28
Average 2006	0.53	0.96	0.64

Tests (#221-247) manipulate structure. In this set of test cases our algorithm performed very well. This was due to the fact that the terms in these test cases had high string similarity, and in the cases where specific terms did not have similar names or comments, our algorithm was able to use structural or semantic features of each ontology to derive the remaining alignments.

	Precision	Recall	F-Measure
Average 2005	0.75	0.86	0.76
Average 2006	0.99	1.00	0.99

Tests (#248-266) randomize the names and comments while manipulating structure. In this set of test cases our algorithm performed very poorly. Since we rely heavily on string similarity we were unable to extract meaningful results from this section.

	Precision	Recall	F-Measure
Average 2005	0.07	0.00	0.00
Average 2006	0.06	0.58	0.11

The last set of tests (#301-304) use ontologies that are adapted from real life ontologies. Since they were not initially created for the purposes of the OAEI library, they give some insight as to how well each algorithm will perform outside of testing. In this set of test cases the set of terms in either ontology never subsumed the other. This means that there were a number of terms within each ontology that were not meant to be aligned. In addition there were several terms that were synonyms of each other. These two factors led to a heavy reliance on our structure and semantic algorithm components, which lead to poor recall and precision.

	Precision	Recall	F-Measure
Average 2005	0.72	0.51	0.55
Average 2006	0.19	0.61	0.28

3 Comments on Results

As Onto-Mapology demonstrates, our algorithm performed very well when names were highly similar, as did many other solutions in the OAEI 2005. Onto-Mapology was able to derive the terms that did not match lexically, as long as there were enough aligned terms to make those associations, given the semantic and structural aspects of our algorithm. Since we used a combination of methods our weaknesses came into effect when: a) names were random or dissimilar; b) comments were random or dissimilar; c) structures of two disjoint objects were identical; d) semantics of two disjoint objects were similar (e.g. same subclass). The test cases were extremely well conceived. They cover a wide variety of cases and also attempt to isolate specific weaknesses within algorithms. They also include real world ontologies which may give indication of how the algorithm will perform in practice.

4 Conclusion

Onto-Mapology is an ontology mapping solution that is both flexible and interactive. Users can choose from a number of matching techniques and apply a single matching technique or a preconfigured combination of matching techniques. Users may also choose our hybrid matching algorithm that brings together several matching techniques across linguistics, structure and semantics. The results of using the hybrid algorithm are discussed in this paper and we have some work to do to improve the performance. The hybrid solution within Onto-Mapology will perform very well as ontologies become more structurally sophisticated and semantically expressive.

References

1. Milo, T., Zohar, S., Using Schema Matching to Simplify Heterogeneous Data Translation, In Proceedings of the International Conference on Very Large Databases (VLDB), 1998
2. Rahm, E., Bernstein, P. A., A Survey of Approaches to Automatic Schema Matching, the VLDB Journal, 2001
3. Zhou, N., A Study on Automatic Ontology Mapping of Categorical Information, 2002
4. Doan, A., Madhavan, J., Domingos, P., Halevy, A., Learning to Map between Ontologies on the Semantic Web, 2002
5. Eclipse.org Home, http://www.eclipse.org/, 2006
6. Noy, N., Musen, M., PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment, In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, 2000
7. McGuinness, D. L., Fikes, R., Rice, J., Wilder, S., An Environment for Merging and Testing Large Ontologies, In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, 2000
8. Ehrig, M., Sure, Y., Ontology Mapping: An Integrated Approach, Accepted for publication at 1st European Semantic Web Symposium, 2004
9. Fensel, D., Hendler, J., Lieberman, H., Whalster, W., *The Semantic Web: Why, What, and How*, MIT Press, 2001
10. SecondString Project Page, http://secondstring.sourceforge.net/, 2006
11. Cohen, W. W., Ravikumar, P., Fienberg, S. E., A Comparison of String Distance Metrics for Name-Matching Tasks, Carnegie Mellon University, 2003
12. White, S., "How to Strike a Match," *Development Cycles*, 2004
13. Lucene Search Engine, http://lucene.apache.org/java/docs/, 2006
14. Pellet, http://www.mindswap.org/2003/pellet/index.shtml, 2006
15. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S., Ontology-Based Integration of Information-A Survey of Existing Approaches, In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), 2001
16. De Bruijn, J., et. al., “State-of-the-art Survey on Ontology Merging and Aligning,” Digital Enterprise Research Institute, University Innsbruck, 2004
17. Kalfoglou, Y., Schorlemmer M., Ontology Mapping: The State of the Art, Dagstuhl Seminar Proceedings, Semantic Interoperability and Integration, 2005

NLM Anatomical Ontology Alignment System
Results of the 2006 Ontology Alignment Contest

Songmao Zhang[1], Ph.D., Olivier Bodenreider[2]

[1] Institute of Mathematics, Academy of Mathematics and Systems Science
Chinese Academy of Sciences, Beijing, P. R. China
[2] U.S. National Library of Medicine, NIH, Bethesda, Maryland, USA
[1] `smzhang@math.ac.cn`, [2] `olivier@nlm.nih.gov`

Abstract. An ontology is a formal representation of a domain modeling the entities in the domain and their relations. When a domain is represented by multiple ontologies, there is need for creating mappings among these ontologies in order to facilitate the integration of data annotated with and reasoning across these ontologies. The objective of this paper is to present our experience in aligning two large anatomical ontologies and to reflect on some of the issues and challenges encountered along the way. The anatomical ontologies under investigation are the Foundational Model of Anatomy (FMA) and GALEN. Our approach to aligning concepts is automatic, rule-based, and operates at the schema level, generating mostly point-to-point mappings. It uses a combination of domain-specific lexical techniques and structural and semantic techniques (to validate the mappings suggested lexically). It also takes advantage of domain-specific knowledge (lexical knowledge from external resources such as the Unified Medical Language System, as well as knowledge augmentation and inference techniques). Overall, the lexical alignment followed by structural validation identified 3,029 pairs of equivalent concepts in the FMA and GALEN, accounting for about 4% of all FMA concepts and 32% of all GALEN concepts.

1 Presentation of the system

Over the past four years, as part of the Medical Ontology Research project at the U.S. National Library of Medicine, we have developed domain knowledge-based techniques for aligning large anatomical ontologies, with the objective of exploring approaches to aligning representations of anatomy differing in formalism, structure, and domain coverage. We started by aligning concepts point-to-point in two large ontologies of human anatomy, using lexical and structural techniques [1]. We later tested these techniques on other pairs of anatomical ontologies, both within and across species [2, 3]. We also investigated the complex alignment of groups of concepts [4] and that of relationships [5]. Finally, we investigated the possibility of deriving the indirect alignment of two ontologies through their direct alignment to a reference ontology [6]. The objective of this paper is to present our experience in aligning two large anatomical ontologies point-to-point and to reflect on some of the issues and challenges encountered along the way. In particular, we want to show the importance of domain-specific knowledge in our alignment strategies. Many features of our system are specific to the domain of anatomy, which is why ‘anatomy’ is the only OAEI data set to which we applied our methods.

1.1 State, purpose, general statement

The approach to aligning anatomical ontologies presented here is automatic, rule-based, and operates at the schema level, generating point-to-point mappings. It uses a combination of domain-specific lexical techniques (to map entities at the element, not instance level) and structural and semantic techniques (to validate the mappings suggested lexically). It also takes advantage of domain-specific knowledge (lexical knowledge from external resources such as the Unified Medical Language System® (UMLS®) [7], as well as knowledge augmentation and inference techniques).

The many ontology alignment systems available include PROMPT [8], CUPID [9], FCA-Merge [10], HCONE-Merge [11], and GLUE [12]. With AnchorPrompt [13], we share the notion of “anchor” (i.e., a pair of related terms across ontologies, established by lexical similarity in our case) and the use of shared paths between anchors across ontologies to validate the similarity among related terms. Therefore, AnchorPrompt is undoubtedly the system to which our approach is the most closely related. The major differences between AnchorPrompt and our approach can be summarized as follows. AnchorPrompt creates a sophisticated similarity score based on path length and other features. In contrast, we use a simpler validation scheme based on paths restricted to combinations of taxonomic and partitive relations, suitable for the anatomical domain. Unlike AnchorPrompt, our approach does not rely on path length and is therefore less sensitive to differences in granularity between ontologies. AnchorPrompts requires ontologies to be accessible in the frame-based Open Knowledge Base Connectivity (OKBC) protocol, while our approach is not specific to any particular formalism.

1.2 Specific techniques used

We identify one-to-one concept mappings between the Foundational Model of Anatomy (FMA) and GALEN using lexical resemblance between concept names and then validate the mappings through shared hierarchical paths among concepts across ontologies.

Lexical alignment

The lexical alignment identifies shared concepts across ontologies based on lexical similarity between concept names. Both preferred concept names and synonyms, if any, are used in the lexical alignment process. Lexical similarity is assessed through exact match and after normalization. The normalization program distributed with the UMLS provides a linguistically-motivated model for lexical resemblance adapted to the specificity of biomedical terms, abstracting away from minor differences in terms including case, hyphen, inflection and word order variations [14].

Concepts exhibiting similarity at the lexical level across ontologies are called anchors, as they are going to be used as reference concepts in the structural validation and for comparing associative relationship. Additional anchors are identified through synonymy in an external resource: the Unified Medical Language System (UMLS). More specifically, two concepts across ontologies are considered anchors if their names are synonymous in the UMLS Metathesaurus (i.e., if they name the same concept) and if the corresponding concept is in the anatomy domain (i.e., has a semantic type related to *Anatomy*).

Examples of anchors, shown in **Figure 1**, include the concepts *Cardiac valve* in FMA and *Valve in heart* in GALEN, identified as anchor concepts because *Cardiac valve* has *Valve of heart* as a synonym in FMA and *Valve in heart* matches *Valve of heart* after normalization. Additionally, *Fibrous ring of mitral valve* (with synonym *Mitral anulus*) in the FMA and *Mitral ring* in GALEN form an anchor because *Mitral anulus* and *Mitral ring* are synonyms, i.e., they are both names for the concept *Structure of anulus fibrosus of mitral orifice* in the UMLS.

Structural validation

In the structural validation of the lexical alignment, the first step is to acquire the semantic relations explicitly represented in the ontologies. Inter-concept relationships are generally represented by semantic relations <c_1, r, c_2>, where the relationship r links concepts c_1 and c_2. Because they form the backbone of anatomical ontologies and are therefore more likely to be represented consistently across ontologies, hierarchical relationships only are considered at this step. These relationships are *IS-A* and *PART-OF*, along with their inverses *INVERSE-IS-A* and *HAS-PART*, respectively. Having extracted the relations explicitly represented in the ontologies, we then normalize the representation of the relations in each ontology in order to facilitate structural comparisons across ontologies. We first complement the hierarchical relations represented explicitly with their inverses as necessary. Implicit semantic relations are then extracted from concept names (augmentation) and various combinations of hierarchical relations (inference). Augmentation and inference are the two main techniques used to acquire implicit knowledge from the FMA and GALEN. For a detailed analysis of the contribution of each technique, the interested reader is referred to [15].

Complementation. As partial ordering relationships, hierarchical relationships are anti-symmetric. *IS-A* and *PART-OF* have inverse relationships, *INVERSE-IS-A* and *HAS-PART*. Except for *IS-A*, not every relation is represented bidirectionally. For example, <*Right breast proper*, *HAS-PART*, *Right mammary gland*> is explicitly represented in the FMA but its inverse relation is missing. In canonical anatomy, the inverse relations are essentially always valid, although this is not necessarily the case in the real world [16]. For the sole purpose of aligning ontologies, in order to facilitate the comparison of paths between anchors across ontologies, we complement the FMA and GALEN with the inverse relations that are not explicitly represented. For example, we generated the relation <*Right mammary gland,* PART-OF, *Right breast proper*>.

Augmentation attempts to represent with relations knowledge that is otherwise embedded in the concept names. Augmentation is based on linguistic phenomena, such as the reification of partitive relations. In this case, a relation <***P***, PART-OF, ***W***> is created between concepts ***P*** (the part) and ***W*** (the whole) from a relation <***P***, IS-A, ***Part of W***>, where the concept ***Part of W*** reifies, i.e., embeds in its name, the PART-OF relationships to ***W***. For example, <*Neck of femur,* PART-OF, *Joint*> was added from the relation <*Neck of femur,* IS-A, *Component of joint*>, where the concept *Component of joint* reifies a specialized PART-OF relationship. Examples of augmentation based on other linguistic phenomena include <*Sweat gland,* IS-A, *Gland*> (from the concept name *Sweat gland*) and <*Extensor muscle of leg,* PART-OF, *Leg*> (from the concept name *Extensor muscle of leg*). The semantics of nominal modification generally corresponds to subsumption (e.g., the head noun *gland* modified by *sweat* is a hypernym of *gland*). In contrast, the semantics of prepositional clauses introduces by *of* is not necessarily a partitive relation (e.g., *glass of wine* is not part of *wine*). Here, domain knowledge was required to assess what relations can be automatically extracted with high accuracy in the particular context of anatomical terms. We determined that partitive relations could be accurately created from prepositional clauses introduced by *of* in anatomical terms containing no other prepositions.

Inference generates additional semantic relations by applying inference rules to the existing relations in order to facilitate the comparison of paths between anchors across ontologies. These inference rules, specific to this alignment, represent limited reasoning along the PART-OF hierarchy, generating a partitive relation between a specialized part and the whole or between a part and a more generic whole. For example, <*First tarsometatarsal joint,* PART-OF, *Foot*> was inferred from the relations <*First tarsometatarsal joint,* IS-A, *Joint of foot*> and <*Joint of foot,* PART-OF, *Foot*>. Analogously, <*Interphalangeal joint of thumb,* PART-OF, *Finger*> was inferred from the relations <*Interphalangeal joint of thumb,* PART-OF, *Thumb*> and < *Thumb,* IS-A, *Finger*>.

With these explicit and implicit semantic relations, the structural validation identifies structural similarity and conflicts among anchors across ontologies. Structural similarity, used as **positive structural evidence**, is defined by the presence of common hierarchical paths among anchors across ontologies, e.g., <c_1, PART-OF, c_2> in one ontology and <c_1', PART-OF, c_2'> in another where $\{c_1, c_1'\}$ and $\{c_2, c_2'\}$ are anchors across ontologies[1]. The anchor concepts *Cardiac valve* in FMA and *Valve in heart* in GALEN, presented earlier, received positive structural evidence because they share hierarchical paths to some of the other anchors across ontologies. For example, as illustrated in **Figure 1**, *Cardiac valve* is related to *Heart* (*PART-OF*), to *Mitral valve* (*INVERSE-IS-A*) and to *Mitral ring* (*HAS-PART*).

Conflicts, on the other hand, are used as **negative structural evidence**. The first type of conflict is defined by the existence of hierarchical paths between the same anchors across ontologies going in opposite directions, e.g., <c_1, PART-OF, c_2> in one ontology and <c_1', HAS-PART, c_2'> in another. The second type of conflict is based on the disjointness of top-level categories across ontologies (i.e., semantic constraints).

---

[1] The transitive closure of hierarchical relation greatly facilitates paths comparison across ontologies, because complex paths between anchors are represented by a single relation.

For example, *Point* in FMA is a kind of *Dimensional entity*, while *Pointing* in GALEN is a *Voluntary movement of upper extremity*, which is a *Process*. *Dimensional entity* and *Process* being disjoint top-level categories, the two concepts *Point* in the FMA and *Pointing* in GALEN are considered semantically distinct, which prevents them from being aligned although they are lexically equivalent (after normalization).

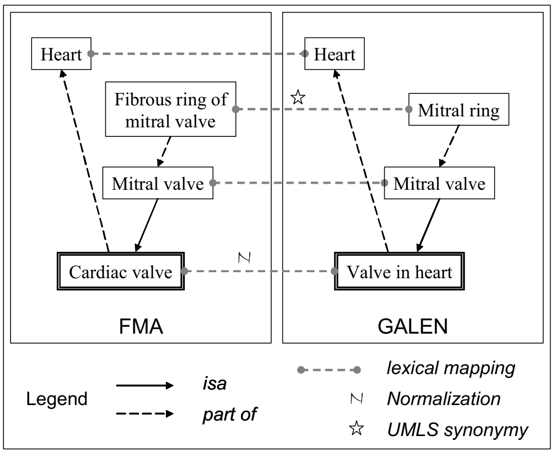


Figure 1. Structural validation following lexical alignment

1.3 Adaptations made for the evaluation

Extracting FMA and GALEN from OWL Full. In previous alignment experiments involving the FMA and GALEN, we purposely stayed away form any particular formalism in order to avoid distorting the source ontologies during the conversion process. Instead, we simply extracted <concept, relationship, concept> triples from the two ontologies in their original formalism (frames in Protégé for the FMA and the description logic language GRAIL for GALEN) using the Application Programming Interface provided with the ontology. In the OAEI experiment, the FMA and GALEN were converted into OWL Full by the organizers. Since our alignment approach is not designed to specifically take advantage of the OWL formalism, we again extracted concept names and <concept, relationship, concept> triples from the class definitions in OWL.

More specifically, for the FMA, we used ***rdf:ID*** to identify concepts, ***rdf:label*** and ***Preferred_name*** to acquire concept names, and ***rdfs:subClassOf*** to acquire taxonomic relations. The various partitive relations represented in the FMA (e.g., *part_of*, *constitutional_part_of*, *regional_part_of*) were acquired using the corresponding properties. An inverse property is specified for 29 properties in the OWL file. The other properties were not used in the alignment.

Because of the deep nesting exhibited by the GALEN file in OWL, we performed the extraction not from the OWL file itself, but from the *.pont* and *.pins* files resulting from its conversion into the CLISP format by the OWL plugin in Protégé. The *.pins* file contains all the information we needed and is the only one we ended up using. (This file is available for download at: http://mor.nlm.nih.gov/pubs/supp/2006-oaei-sz/OpenGALEN.pins). We used ***rdf:ID*** to both identify concepts and acquire concept names (after tokenization), and ***rdfs:subClassOf*** to acquire taxonomic relations. The various partitive relations represented in GALEN (e.g., *isPartOf, isComponentOf, IsSurfaceDivisionOf*) were acquired using the corresponding properties. No inverse properties are specified in the OWL file. Inverse properties were added manually in 13 cases for alignment purposes. The other properties were not used in the alignment.

Grouping fine-grained partitive relationshipss. For alignment purpose in this study, we consider as only one *PART-OF* relationship (with *HAS-PART* as its inverse) the various kinds of partitive relationships present in the FMA (e.g., *part_of*, *constitutional_part_of*, *regional_part_of*) and GALEN (e.g., *isPartOf, isComponentOf, IsSurfaceDivisionOf*).

Adding disjointness axioms between top-level classes. In order to identify semantic mismatches, we added pairwise disjointness axioms between their top-level classes across the two ontologies. For example, as shown earlier, because we define *Dimensional entity* in the FMA and *Process* in GALEN as being disjoint top-level categories, *Point* in FMA – a kind of *Dimensional entity* – and *Pointing* in GALEN – a kind of *Process* – are prevented from being aligned although they are lexically equivalent after normalization.

1.5 Link to the set of provided alignments (in align format)

The result of our alignment for the 'anatomy' data set is available at: http://mor.nlm.nih.gov/pubs/supp/2006-oaei-sz/Zhang&Bodenreider.rdf in the format specified by the OAEI organizers at: http://oaei.ontologymatching.org/2006.

2 Results

The only data set for which we report results is 'anatomy'. Overall, the lexical alignment followed by structural validation identified 3,029 pairs of equivalent concepts in the FMA and GALEN, accounting for about 4% of all FMA concepts and 32% of all GALEN concepts.

Acquiring concept names and relations for the FMA and GALEN. The main characteristics of the data sets under investigation are listed in **Table 1**, including the number of classes, concept names, and types of partitive relationships. The number of *IS-A* and partitive relations extracted from the OWL file and generated by complementation, augmentation and inference is shown in **Table 2**. Not surprisingly, in both ontologies, a majority of relations come from inference, which performs similarly to a transitive closure of the hierarchical relations. Also listed in **Table 2** is the small number of relations removed from the ontologies because they create cycles.

Table 1. Main characteristics of the FMA and GALEN

#	FMA	GALEN
Concepts	72,560	9,566
Synonyms	44,597	0
Anonymous concepts	0	1,035
Part-of relationships	7	8
Has-part relationships	7	8
Inter-concept associative relationships	67	13
Datatype properties defined in the owl file	19	2

Table 2. Number of relations in the FMA and GALEN

Types of relations	FMA	GALEN
Explicitly represented is-a	72,560	18,091
Explicitly represented partitive relations	101,161	12,830
Explicitly represented associative relations	48,804	8,341
Complemented inverse-isa	72,560	18,091
Complemented partitive relations	3,561	4,364
Complemented associative relations	11,697	762
Removed because of cycles	- 40	- 2
Augmented	169,378	29,780
Inferred	5,169,034	243,436
Total	5,648,715	335,693

Table 3. Results of structural validation for the FMA-GALEN alignment

Structural evidence		3,132 anchors		
Positive evidence	Shared paths to other anchors(same type)	2056	3029	96.7%
	Shared paths to other anchors (“compatible”)	973		
Negative evidence	Conflicting paths to other anchors	9	22	0.7%
	Semantic disjointness	13		
No evidence	No paths to other anchors	61	81	2.6%
	No shared paths to other anchors	20		

Lexical alignment. 3,132 matching anchor concepts were identified lexically, accounting for about 4.3% of the FMA concepts and 32.7% of GALEN concepts. Of these, 378 anchors were identified through UMLS synonymy.

Structural validation. For the 3,132 anchors, the presence of paths to other anchors is searched in both ontologies, as well as the existence of conflicts is assessed. This information is used for the structural validation of the alignment, the results of which are summarized in **Table 3**.

Anchors with positive structural evidence. 96.7% of the anchors receive positive evidence, most of them sharing hierarchical paths of the same type (e.g., *Cardiac valve* in FMA and *Valve in heart* in GALEN, presented earlier). An example of shared "compatible" hierarchical relations is the anchor *Pelvic fascia*. In both ontologies, this concept is linked to *Visceral pelvic fascia*, but, although going in the same direction, the relationship is *INVERSE-ISA* in GALEN and *HAS-PART* in FMA. For alignment purposes, sharing compatible hierarchical relations is deemed a sufficient condition. Anchors with positive structural evidence are presented in our result file with "=" as value for "relation" and 1.0 as value for "measure".

Anchors with negative structural evidence. About 0.7% of the anchors represent conflicts between the two ontologies. For example, the relationship between the anchors *Apex of bladder* and *Urinary bladder* is *PART-OF* in GALEN but *HAS-PART* in FMA. Another type of conflict is represented by the semantic incompatibility between *Point* (the dimensional entity) in the FMA and *Pointing* (the process) in GALEN presented earlier. Anchors with negative structural evidence are presented in our result file with "!=" as value for "relation" and 1.0 as value for "measure".

Anchors with no structural evidence. 2.6% of the anchors do not receive any structural evidence. For example, although linked to *Body Cavity (is-a)* and *Peritoneal Cavity (inverse-isa)* in GALEN, *Serous Cavity* has no connections to other anchors in FMA. The absence of *any* paths to other anchors represents about two thirds of the cases. The remaining cases correspond to the absence of *shared* paths to other anchors across ontologies. For example, although *Eyebrow* is linked to two anchors in FMA (e.g., *Set of hairs*), and two in GALEN (e.g., *Face*), none of these paths are shared across ontologies. Anchors with no structural evidence are presented in our result file with "=" as value for "relation" and 0.5 as value for "measure".

3 General comments

3.1 Comments on the results

Overall, the results we obtained on this 'anatomy' data set are essentially similar to the results obtained in previous experiments from slightly different versions of these two large anatomical ontologies [1]. The main difference is that, in the past, we mapped all of GALEN – not just its anatomy subset – to the FMA, leading to increased ambiguity between anatomical and non anatomical concept names. In contrast to what most teams reported at OAEI 2005, for us, this 'anatomy' data set was actually slightly *less* complex than what we are used to.

3.2 Discussions on the way to improve the proposed system

The strengths and weaknesses of our system have been analyzed in previous papers [17]. The major difference with other systems is that we take advantage of domain knowledge throughout the mapping process. For example, we use specific tools and resources, including normalization techniques developed for biomedical terms and synonyms from the Unified Medical Language Systems. We also developed techniques specific to the anatomical ontologies under investigation in order to represent explicitly relations implicitly present in these ontologies. These additional synonyms and relations increase the chances of identifying matches both at the lexical and structural level. Conversely, because most of these techniques are specific to anatomical ontologies, our system is unlikely to perform well on other types of ontologies. Similar techniques would have to be developed for other domains in order to obtain similar levels of performance. Finally, specifying disjointness between top-level categories was more useful in previous experiments when all of GALEN was aligned with the FMA. For example, because we define *Anatomical structure* in the FMA and *Inert solid structure* in GALEN as being disjoint top-level categories, *Nail* in FMA – a kind of *Anatomical structure* – and *Nail* in GALEN – a kind of *Surgical fixation device* – were prevented from being aligned although they are lexically identical.

3.3 Comments on the OAEI 2006 procedure

In our opinion, one of the most controversial aspects of the OAEI procedure is the decision of the organizers to convert the anatomical data sets under investigation – the FMA and GALEN – from their original formalism to OWL Full. On the one hand, the intuition is that having to deal with only one formalism will make it easier for participants to align these two massive ontologies. On the other, the result of the conversion is dependent on the original modeling. The FMA's native environment is frame-based and makes heavy use of metaclasses [18], while GALEN's original formalism is based on the description logic language GRAIL – the GALEN Representation and Integration Language [19]. The mechanical conversion of these two ontologies to OWL Full is therefore likely to result in differing representations, e.g., because of the difference in the use of metaclasses and instances.

In practice, the OAEI participants cannot reliably take advantage of such features of the underlying representation for identifying equivalent entities across ontologies. This issue was illustrated by the example given by Kalfoglou & Hu in their attempt to align the FMA and GALEN in the 2005 OAEI campaign [20]. They specifically question the mapping established by [17] between *Pancreas* in FMA and *Pancreas* in GALEN on the basis that it "is defined as a class in FMA … whereas in GALEN (OpenGALEN) [it is defined] as an instance of [the] class "Body Cavity Anatomy"". Looking beyond formalism, it is quite clear to domain experts that the entities referred to by *Pancreas* in the FMA and GALEN are equivalent. For example, in both ontologies, they are defined as a kind of organ and have the same parts, including exocrine and endocrine pancreas, and head, neck, body and tail of pancreas. The containment relations are slightly different in the FMA (retroperitoneal space) and GALEN (abdominal cavity), but compatible for alignment purposes.

As shown recently, converting the entire FMA to OWL is a difficult exercise because it requires some understanding of both the modeling strategy and the domain, which is beyond what can be expected from the mechanical translation provided by the "export to OWL" function in Protégé [21]. However, the mechanical conversion was sufficient for us to reliably extract the information we needed (namely, concept names and taxonomic and partitive relations). Our point here is converting these two ontologies to OWL Full is an overkill and gives the false impression that the formalism can be relied upon.

3.4 Comments on the OAEI 2006 test cases

In 2005, only two teams participated in the anatomy challenge [20, 22]. Their reports essentially outline the difficulties encountered along the way, including the large size of the anatomical ontologies and the transformation of both ontologies from their native format into OWL Full. Hopefully, more participants will have aligned these two large ontologies in 2006. While anatomical ontologies represent good data sets for evaluating alignment strategies, important issues – beyond size and formalism – remain the absence of a gold standard alignment and the need for domain knowledge.

3.5 Comments on the OAEI 2006 measures

While the format required for reporting results is adequate for point-to-point matches across ontologies, it is not clear how other kinds of mappings could be reported. For example, although not reported here, our system can identify *group-to-group matches* [4] and so do other systems such as AnchorPrompt [13]. Along the same lines, one important feature of our system is to identify *non-matches*, i.e., concepts exhibiting resemblance at the lexical level, but semantic differences. Such anchors with negative structural evidence are presented in our result file with "!=" as value for "relation" and 1.0 as value for "measure". However, an alternative representation for such non-matches would be to use "=" as value for "relation" and a negative value (e.g., -1.0) for "measure". In this case, the meaning of the value parameter would be somewhat equivalent to that of a correlation coefficient.

In the absence of a reference mapping, it is difficult to evaluate alignment systems on this data set. As a biomedical terminology integration resource, the Unified Medical Language System (UMLS) could provide some elements for evaluating the mappings. However, unfortunately, neither the FMA nor GALEN is part of the UMLS. Moreover, the framework for representing anatomical entities in SNOMED CT – one of the source vocabularies in the UMLS – leads to multiple mappings for many anatomical terms (ambiguity) [3]. In previous work, we analyzed the mapping results produced automatically by several systems [17]. Most mappings were identified by the two systems under investigation, which was reassuring. Analyzing the differences, i.e., mappings identified by one system but not by the other, provided interesting insights about the strengths and weaknesses of each system. Cross-validation among the various systems in competition in the OAEI campaign, would not provide a definitive evaluation metric, but could help system developers analyze the mappings specific to their systems and those they missed. The idea of evaluating the mappings in the context of an application suggested by [20] is very interesting, but difficult to realize. Since ontology alignment is key to the Semantic Web, a collaboration with the W3C Semantic Web Health Care and Life Sciences Interest Group could provide the context for such an evaluation.

4 Conclusion

Aligning large anatomical ontologies has generated a lot of interest in the biomedical and computer science communities and successful methods will contribute to create the biomedical component of the Semantic Web. Despite promising advances, aligning anatomical ontologies remains challenging, because of the sheer size of these resources and the need for domain knowledge. Differences in modeling and representation formalism are also an issue, as well as the lack of a reference mapping for evaluation purposes. Competitive evaluation campaigns such as OAEI represent for ontology alignment the same kind of driving force BioCreAtIvE has been for information extraction in biology [23].

Acknowledgments. This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM), and by the Natural Science Foundation of China (No.60496324), the National Key Research and Development Program of China (Grant No. 2002CB312004), the Knowledge Innovation Program of the Chinese Academy of Sciences, MADIS of the Chinese Academy of Sciences, and Key Laboratory of Multimedia and Intelligent Software at Beijing University of Technology.

References

1. Zhang, S., Bodenreider, O.: Aligning representations of anatomy using lexical and structural methods. Proc AMIA Symp (2003) 753-757
2. Bodenreider, O., Hayamizu, T.F., Ringwald, M., de Coronado, S., Zhang, S.: Of mice and men: Aligning mouse and human anatomies. Proc AMIA Symp (2005) 61-65
3. Bodenreider, O., Zhang, S.: Comparing the representation of anatomy in the FMA and SNOMED CT. Proc AMIA Symp (2006) (in press)
4. Zhang, S., Bodenreider, O.: Aligning anatomical ontologies: The role of complex structural rules. Proceedings of the 18th International Conference on Systems Research, Informatics and Cybernetics (InterSymp 2006) (2006) (in press)
5. Zhang, S., Bodenreider, O.: Comparing associative relationships among equivalent concepts across ontologies. Medinfo (2004) 459-463
6. Zhang, S., Bodenreider, O.: Alignment of multiple ontologies of anatomy: Deriving indirect mappings from direct mappings to a reference. Proc AMIA Symp (2005) 864-868
7. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res **32 Database issue** (2004) D267-270
8. Noy, N.F., Musen, M.A.: PROMPT: algorithm and tool for automated ontology merging and alignment. Proceedings of AAAI (2000) 450-455

9. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching using Cupid. Proceedings of 27th International Conference on Very Large Data Bases (2001) 49–58
10. Stumme, G., Maedche, A.: FCA-Merge: Bottom-up merging of ontologies. Prceedings of the 7th International Conference on Artificial Intelligence (IJCAI '01) (2001) 225-230
11. Kotis, K., Vouros, G.A., Stergiou, K.: Towards automatic merging of domain ontologies: The HCONE-merge approach. Web Semantics: Science, Services and Agents on the World Wide Web **4** (2006) 60-79
12. Doan, A., Madhavan, J., Domingos, P., Halevy, A.Y.: Ontology matching: A machine learning experience. In: Staab, S., Studer, R. (eds.): Handbook on Ontologies. Springer-Verlag (2004) 385-403
13. Noy, N.F.: Tools for mapping and merging ontologies. In: Staab, S., Studer, R. (eds.): Handbook on Ontologies. Springer-Verlag (2004) 365-384
14. McCray, A.T., Srinivasan, S., Browne, A.C.: Lexical methods for managing variation in biomedical terminologies. Proc Annu Symp Comput Appl Med Care (1994) 235-239
15. Zhang, S., Bodenreider, O.: Investigating implicit knowledge in ontologies with application to the anatomical domain. In: Altman, R.B., Dunker, A.K., Hunter, L., Jung, T.A., Klein, T.E. (eds.): Pacific Symposium on Biocomputing 2004. World Scientific (2004) 250-261
16. Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A.L., Rosse, C.: Relations in biomedical ontologies. Genome Biol **6** (2005) R46
17. Zhang, S., Mork, P., Bodenreider, O.: Lessons learned from aligning two representations of anatomy. In: Hahn, U., Schulz, S., Cornet, R. (eds.): Proceedings of the First International Workshop on Formal Biomedical Knowledge Representation (KR-MED 2004) (2004) 102-108
18. Noy, N.F., Musen, M.A., Mejino, J.J.L.V., Rosse, C.: Pushing the envelope: challenges in a frame-based representation of human anatomy. Data & Knowledge Engineering **48** (2004) 335-359
19. Rector, A.L., Bechhofer, S., Goble, C.A., Horrocks, I., Nowlan, W.A., Solomon, W.D.: The GRAIL concept modelling language for medical terminology. Artif Intell Med **9** (1997) 139-171
20. Kalfoglou, Y., Hu, B.: CROSI Mapping System (CMS) - Result of the 2005 Ontology Alignment Contest. In: Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H. (eds.): Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies (2005)
21. Golbreich, C., Zhang, S., Bodenreider, O.: The foundational model of anatomy in OWL: Experience and perspectives. Web Semantics: Science, Services and Agents on the World Wide Web **4** (2006) 181-195
22. Jian, N., Hu, W., Cheng, G., Qu, Y.: FalconAO: Aligning ontologies with Falcon. In: Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H. (eds.): Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies (2005)
23. Hirschman, L., Yeh, A., Blaschke, C., Valencia, A.: Overview of BioCreAtIvE: critical assessment of information extraction for biology. BMC bioinformatics [electronic resource] **6 Suppl 1** (2005) S1

OWL-CtxMatch in the OAEI 2006 alignment contest

Sławomir Niedbała

Institute of Computer Science, Silesian University of Technology, Gliwice, Poland
`Slawomir.Niedbala@polsl.pl`

Abstract. The OWL-CtxMatch algorithm is one of the latest developments in the domain of ontology matching. It is next solution that represents a new family of algorithms performing so-called "semantic matching". The paper presents the results of OWL-CtxMatch obtained during its evaluation in the 2006 Campaign organized by Ontology Alignment Evaluation Initiative. The results have been preceded by a brief description of the algorithm and its implementation and completed with some conclusions.

1 Presentation of the system

1.1 State, purpose, general statement

The OWL-CtxMatch algorithm has been designed to match OWL DL ontologies [8]. It is an OWL-specialized version of the CtxMatch algorithm [2], which has been declared as a general algorithm designed to "discover semantic relationships across distinct and autonomous generic structures" [3]. The main requirement it imposes on the structures being matched is the necessity of labelling them with the natural language. The unique feature that both algorithms offer is that they perform so-called "semantic matching" [4] and as a result are able to recognize a broad range of relationships between matched entities, i.e. not only equivalence but also subsumption, disjointness and intersection.

This ability proved to be very useful in the domain of web services discovery [1], especially in the so-called "matchmaking" of web services by means of OWL-S Service Profiles [5]. There are number of algorithms in this domain, e.g. this implemented in OWL-S/UDDI Matchmaker [10], that require knowing exactly what kind of relationship holds between two or more given OWL entities. In this context particularly interesting are two kinds of relationships, i.e. equivalence and subsumption. However, the aforementioned matchmaking algorithms are not prepared for the problem of ontological heterogeneity. Their existing implementations simply ignore the fact that OWL-S Service Profiles might have been described by means of OWL classes and properties defined in different, unrelated ontologies. As a result, these implementations are not able to matchmake described in such a way web services, because they do not recognize hidden relationships that hold between unrelated OWL entities.

Recently there has been proposed a solution to this problem [7] that depends on the idea of adding in to existing tools a supplementary module that matches unrelated OWL entities on demand. The module is expected to be able to interpret ontologies mainly on the basis of their terminological parts, because there is no guarantee that these ontologies will provide sufficient set of instances. On the other hand it is expected that the entities of given ontologies will have been named with commonly used words, not with some unintelligible symbols or randomly generated texts. With this motivation in mind there has been developed the OWL-CtxMatch algorithm, which has already been used in one of the latest OWL-S matchmakers called Cobra1Matchmaker [9].

1.2 Specific techniques used

OWL-CtxMatch similarly as CtxMatch realizes matching as a series of computations of relationships in which each computation is performed for every single pair of unfamiliar entities coming from both given structures. Each mapping is computed here in two steps. At first one there are built internal representations of both entities, by means of which the algorithm stores recognized interpretations. These interpretations are defined in the form of appropriate logical formulas. In OWL-CtxMatch there have been proposed description logics formulas, which are more expressive than propositional logics formulas used in original CtxMatch. The second step of finding single mapping amounts to computing what relationship holds between particular entities on the basis of their internal representations. Since OWL-CtxMatch uses description logics formulas, this step in practice is realized by an external DL reasoner that initially merges both sets of formulas into one model, performs classification of it and finally determines what kind of relationship holds between counterparts for the particular entities.

As it has already been depicted in the paper [2] the essential impact on the way how in the CtxMatch–based algorithms the given entities are interpreted has three levels of available knowledge. They are: lexical knowledge that defines meanings of the words used in the entities labels, domain knowledge that provides relations holding between senses associated with the given entities, and structural knowledge that in case of OWL-CtxMatch is simply interpretation of the given OWL models. The current version of the algorithm uses solely WordNet dictionary [6] as a source of both lexical and domain knowledge and therefore is limited to the English language only. However, the algorithm can be easily modified in order to use other thesauri. Furthermore there is an intention to develop a new version of the algorithm that will be able to use other kinds of sources, e.g. general knowledge or reference ontologies like it has been described in the proposal of the CtxMatch2 algorithm [11].

Since in the OWL structures there can be distinguished two primary elements, that is OWL classes (concepts) and OWL properties (roles), the OWL-CtxMatch algorithm required designing two variants of matching procedures. In both cases the general idea of finding a mapping is the same, however matching of OWL properties is more complicated and, in effect, requires performing double quantity of computations while finding mappings between OWL classes is performed in an original one-step way. Another issue appeared during matching of OWL ontologies that needed to be resolved in OWL-CtxMatch was the problem of how to find mappings for classes and properties whose definitions are mutually dependent. For this reason the current version of the algorithm has simplified the issue by ignoring built-in `rdfs:domain` and `rdfs:range` properties. It means that using any of these restrictions will not affect the matching process at all. As a result definitions of OWL properties are never dependant on any OWL classes and therefore OWL-CtxMatch matches properties at first and classes afterwards, where it can use already found mappings between properties, on which given classes depend.

1.3 Adaptations made for the evaluation

Since the OWL-CtxMatch generates mappings in its internal form, these mappings have to be converted to the official alignment format for the purpose of OAEI 2006 Campaign. This task, however, gave rise to another problem, namely how to perform a proper conversion so the algorithm results would be well comparable with others. Assuming that all of the internal mappings are transformed to the alignment format, there would be produced very large resultant alignments during these conversions, because for each pair of entities coming from different ontologies there would exist one mapping tuple. Thus it would drastically deteriorate precision results achieved by the algorithm without getting any gain in recall values. For this reason in the current approach during the conversion there are transformed only those mappings, in which recognized relationship is equivalence. These filtered mappings are then written in the files named “OCM.rdf” (after the first letters of Owl-CtxMatch).

Apart from them there are also created supplementary files named “OCM_full.rdf”, in which there are additionally allowed mappings having subsumption relations. It is worth mentioning that although these files are extended with the “_full” suffix, they actually do not contain all the mappings. It probably would be more appropriate to name them “OCM_extended.rdf”, what might be reconsidered in the next versions of the application. Finally it must be emphasized that these supplementary files are not the part of submission and they serve for informative purposes only.

1.4 Link to the system and parameters file

The OWL-CtxMatch implementation along with all its required libraries and information how to run the application is available at the following URL:

http://www-zo.iinf.polsl.gliwice.pl/~niedbyk/oaei2006/

1.5 Link to the set of provided alignments (in align format)

The alignments produced by the OWL-CtxMatch are available on the same web site as mentioned before, i.e. they can be found at:

http://www-zo.iinf.polsl.gliwice.pl/~niedbyk/oaei2006/

2 Results

2.1 Benchmark tests

The first part of the benchmark tests (#1xx) was quite easy for OWL-CTXMATCH. The algorithm found all the required relationships, however apart from them it also discovered some additional mappings that caused the algorithm's precision has deteriorated slightly. These mapping appeared to be an effect of the conflicts between structural knowledge gained from OWL ontologies and domain knowledge provided by WordNet. For instance, the reference ontology defines an OWL class named `Book` as a subclass of `Reference`, while in the WordNet one of the senses associated with `reference` is a hyponym of (a kind of) one of the senses associated with the `book`. As a result OWL-CTXMATCH deduces wrongly that both meanings are equivalent.

Other worth considering aspects of initial part of benchmark tests are their durations. Since the actual matching in OWL-CTXMATCH is preceded with classifications of both input ontologies, the duration of all the computations is highly dependant on how long these classifications last. It is very noticeable that in case of matching the reference ontology with the wine ontology (test #102) the whole process took the algorithm as many as five hours. The reason was that the preliminary classification of the wine ontology was very time-consuming. In other cases matchings lasted considerably shorter, however they still required about six, seven minutes. Such long durations are connected with the fact that the algorithm computes all the mappings for each pair of entities coming from different ontologies. For instance it had to recognize more than three thousand of relationships in case of test #101 only.

The second part of the benchmark tests (#2xx) confirmed that the OWL-CTXMATCH algorithm is strictly dependent on correct labelling ontology entities with the natural language words. It can be observed on the basis of the tests, in which recall results are equal or almost equal to zero. Such values have been obtained in tests #201, #202, #206, #207, #210 and also in #248-#266. In these tests labels of ontology entities have been either replaced by random texts or by their French translations (these non-English translations are not understandable by OWL-CTXMATCH, which is a WordNet-based system).

Unexpectedly the algorithm achieved also poor recall results in the tests #205 and #209, in which synonyms have been used. It proves that WordNet, which currently fulfils a function of the source of lexical and domain knowledge, definitely needs to be aided with knowledge coming from other sources. It is also worth mentioning that in those two tests (#205 and #209) the matching durations have increased rapidly. It is fully understandable, because each synonym brings by means of its label new words and in case of polysemies their additional senses to internal formulas what causes OWL-CTXMATCH to operate on the bigger internal representations.

The next alterations made in test ontologies, like removing instances (#224), modifying concepts hierarchy (#221-#223) or removing properties (#225, #228) etc. did not seem to affect OWL-CTXMATCH matching abilities or affected them in a small extent. Furthermore, the good values of precision achieved by the algorithm in almost all tests of the second group confirmed the author in the conviction that OWL-CTXMATCH can be useful not only in matchmaking of Web Services but also in other applications that require sophisticated ontology matching.

The last part of benchmark tests (#3xx) consisted of real ontologies and proved that the results achieved in previous two parts were not the outcome of special tuning of the algorithm to the their sets of synthetic ontologies. The obtained precisions of about 90% and the recalls of about 50% allow describing the OWL-CTXMATCH algorithm as a solid and robust solution in the domain of ontology matching.

2.2 Directory and conference tests

The OWL-CTXMATCH algorithm managed to perform all the directory tests without any serious problems. Since they are blind tests and there are not reference mappings available for this moment, it is difficult to comment obtained results. The similar situation of lacking reference mappings arises in reference to the conference tests, however in this case some of the ontologies have not been matched.

Firstly two out of ten conference ontologies (`confious.owl`, `OpenConf.owl`) turned out to be in OWL Full. It generally was impossible for OWL-CTXMATCH to deal with them, since the algorithm is designed to matching OWL DL ontologies. Furthermore, the algorithm had unexpected problems with interpreting IASTED ontology (`iasted.owl`), even it was in OWL DL. The reason for this was the fact that the Pellet reasoner [12], which is used by the current implementation of OWL-CTXMATCH, could not classify this ontology in a reasonable time. However, it must be said that there was not used the latest stable version of Pellet, but its nightly build from 2006-03-30/31. Since it is required for every ontology to be classified before actual matching begins, OWL-CTXMATCH never could even have started to match anything with IASTED ontology. Pellet is also to blame for the fact that OWL-CTXMATCH could not match `sigkdd.owl` with `cmt.owl`. This is the only case excluding IASTED ontology when the OWL-CTXMATCH algorithm could not handle matching two proper OWL DL ontologies.

3 Conclusion

In the paper there has been presented the results of the OWL-CTXMATCH algorithm obtained during its evaluation in the OAEI 2006 Campaign. These results allow stating that the examined algorithm is a reliable solution for matching OWL DL ontologies. However, the evaluation that has been carried out did not take into considerations all the aspects of the examined solution, in particular, its special ability to perform semantic matching. Since almost all of the provided reference mappings have only equivalence relationships, it was impossible to assess other kinds of mappings recognized by OWL-CTXMATCH. Furthermore, the reference set of relationships was prepared based on the assumption that all the examined algorithms will compute one-to-one mappings, while OWL-CTXMATCH generates a set of mapping tuples with the n:m multiplicity. These two essential problems require extending the current evaluation approach so it will be possible to correctly assess results of the algorithms that perform semantic matching. Obviously it will entail alterations in the proposed set of test ontologies as well.

Despite these shortcomings of the evaluation method, the algorithm still needs some further improvements. At first there is a need to refine the methods of obtaining lexical and domain knowledge. As it has already been mentioned, there is a successor of CTXMATCH that allows using reference ontologies as a source of domain knowledge [11]. Other aspect of improvements concerns the algorithm's efficiency. It is currently considered distributing computations of single mappings to other machines and introducing iterative method of matching procedure so it would allow fast pruning the pairs of entities that are definitely not correlated.

References

[1] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web Services Architecture. (2004) http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[2] Bouquet, P., Serafini, L., Zanobini, S.: Semantic coordination: a new approach and an application. In Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.): The Semantic Web - ISWC 2003. Proceedings. Lecture Notes in Computer Science, Vol. 2870. Springer, Berlin Heidelberg (2003) 130–145

[3] Bouquet, P., Serafini, L., Zanobini, S., Benerecetti, M.: An algorithm for semantic coordination. In Proceedings of the Semantic Integration Workshop at the Second International Semantic Web Conference. Sanibel Island, Florida, USA (2003) 21-26.

[4] Giunchiglia, F., Shvaiko, P.: Semantic Matching. The Knowledge Engineering Review, Vol. 18(3). (2003) 265-280

[5] Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., Sycara, K.: Bringing Semantics to Web Services: The OWL-S Approach. In Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition. San Diego, California, USA (2004) 26-42

[6] Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.:. Introduction to WordNet: An On line Lexical Database. International Journal of Lexicography, 3(4), (1990) 235 244.

[7] Niedbała, S.: Towards Semantic Matching of OWL-S Service Profiles. In Proceedings of International Workshop on Control and Information Technology. Ostrava, Czech Republic (2005)

[8] Niedbała, S.: OWL-CTXMATCH: Algorytm semantycznego dopasowywania ontologii OWL (OWL-CTXMATCH: Algorithm for Semantic Matching of OWL Ontologies; in Polish). In Proceedings of 2nd Conference on Bazy Danych: Aplikacje i Systemy. Ustroń, Poland (2006)

[9] Niedbała, S.: Odkrywanie semantycznych usług Web Service w środowisku nieskojarzonych ontologii OWL (Semantic Web Services Discovery in the Environment of Unfamiliar Ontologies; in Polish). In Proceedings of 13th Conference on Sieci Komputerowe. Zakopane, Poland (2006)

[10] Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic Matching of Web Services Capabilities. In Proceedings of the 1st International Semantic Web Conference. Sardinia, Italy (2002) 333-347

[11] Sceffer, S., Serafini, L., Zanobini, S.: Semantic Coordination of Hierarchical Classifications with Attributes. Technical Report no 706, Department of Information and Communication Technology, University of Trento, Italy (2004).

[12] Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A Practical OWL-DL Reasoner. Technical Report 2005-68. University of Maryland, Institute for Advanced Computer Studies, (2005)

Appendix: Raw results

Matrix of results

The following results have been achieved by the OWL-CtxMatch Java 5 implementation that was using the following external libraries: Pellet OWL reasoner (nightly build from 2006-03-30/31) and all its dependants, Jena 2.3 and all its dependants (included in Pellet package), Java WordNet Library 1.3 and its dependant. Thanks to the JWNL library, OWL-CtxMatch was able to communicate with WordNet 2.0, which is also required by the application. The presented below time values were obtained on a dual-processor machine (2 x Intel Xeon 3.06 GHz) with 1 GB RAM and Windows 2003 Server installed on it. The OWL-CtxMatch application was executed in the "Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_08-b03)" with 256 MB of initial heap space and 768 MB of maximum heap space. However, the obtained time values are somewhat approximate, since the reference ontology was initially classified just once (not in every test) and, moreover, the host server was not dedicated to performing ontology matching tasks only. The presented values of precision and recall have been computed by designed for this purpose the "Ontology alignment API and implementation" tool (version 2.4+ from 10-07-2006) recommended by OAEI.

#	Name	Prec.	Rec.	Time
101	Reference alignment	0.96	1.00	00:07:32
102	Irrelevat ontology	NaN	NaN	05:06:25
103	Language generalization	0.92	1.00	00:06:00
104	Language restriction	0.98	1.00	00:06:02
201	No names	1.00	0.01	00:02:59
202	No names, no comments	1.00	0.01	00:02:52
203	No comments	0.97	1.00	00:05:11
204	Naming conventions	0.96	0.90	00:05:14
205	Synonyms	0.85	0.23	00:17:54
206	Translation	1.00	0.23	00:07:17
207		1.00	0.23	00:06:57
208		0.95	0.90	00:05:15
209		0.85	0.23	00:20:59
210		1.00	0.23	00:07:13
221	No specialisation	0.91	0.98	00:04:38
222	Flatenned hierachy	0.93	1.00	00:04:53
223	Expanded hierarchy	0.97	0.96	00:08:59

224	No instance	0.96	1.00	00:04:59
225	No restrictions	0.93	1.00	00:04:04
228	No properties	0.75	1.00	00:01:05
230	Flatenned classes	0.95	1.00	00:04:58
231		0.97	1.00	00:05:06
232		0.99	0.98	00:04:30
233		0.94	0.97	00:01:01
236		0.87	1.00	00:01:04
237		0.98	1.00	00:04:36
238		0.93	0.97	00:08:41
239		0.74	1.00	00:01:07
240		0.85	0.88	00:02:52
241		0.86	0.97	00:00:59
246		0.85	1.00	00:01:08
247		0.81	0.88	00:02:51
248		1.00	0.01	00:02:45
249		1.00	0.01	00:02:46
250		NaN	0.00	00:00:52
251		1.00	0.01	00:03:22
252		1.00	0.01	00:11:17
253		1.00	0.01	00:03:10
254		NaN	0.00	00:01:03
257		NaN	0.00	00:01:04
258		1.00	0.01	00:03:22
259		1.00	0.01	00:10:36
260		0.00	0.00	00:01:11
261		0.00	0.00	00:02:47
262		NaN	0.00	00:01:01
265		0.00	0.00	00:01:12
266		0.00	0.00	00:02:48
301	BibTeX/MIT	0.91	0.52	00:02:29
302	BibTeX/UMBC	0.95	0.40	00:02:16
303	Karlsruhe	0.77	0.41	00:10:59
304	INRIA	0.92	0.63	00:07:03

PRIOR System: Results for OAEI 2006

Ming Mao, Yefei Peng

University of Pittsburgh, Pittsburgh, PA, USA
{mingmao,ypeng}@mail.sis.pitt.edu

Abstract. This paper summarizes the results of PRIOR system, which is an ontology mapping system based on Profile pRopagation and InfOrmation Retrieval techniques, for OAEI 2006 campaign. The PRIOR system exploits both linguistic and structural information to map small ontologies, and integrates Indri search engine to process large ontologies. The preliminary results of the experiments for four tasks (i.e. benchmark, web directories, anatomy and food) are presented. A discussion of the results and future work are given at the end.

1 Presentation of the system

1.1 State, purpose, general statement

The World Wide Web (WWW) makes a large number of digital resources publicly accessible. However, finding relevant information, i.e. searching for digital resources from various sources and manually organizing them for relevance, becomes more and more intractable. Semantic interoperability research is aimed at enabling different information systems to communicate information consistently with the intended meaning. Ontology mapping is one critical mechanism to achieve semantic interoperability.

Different communities have proposed different approaches to ontology mapping. The techniques that have been applied to solve mapping problems include linguistic analysis of terms [5][11], comparison of graphs corresponding to the structures [11], mapping to a common reference ontology [4], use of heuristics that look for specific patterns in the concepts definitions [10][8][12][9], and machine-learning techniques [7][2][3][1].

Our approach begins with the belief that the combination of linguistic analysis and graph theory will lead to successful mapping. It explores information from two perspectives, linguistic and structural, to determine the correspondences that identify similar elements in different ontologies. In an ontology, linguistic information is the descriptive information, such as name (i.e. ID), label, comment and property restriction, of a concept (i.e. class, individual and property). Structural information refers to relationships between concepts in the ontology. Such relationships include hierarchy relation, inverse relation and so on. Since the filed of information retrieval is highly relevant to ontology mapping, we also explore using classic information retrieval method to support the mapping of large ontologies. Figure 1 depicts the architecture of PRIOR system. The details of the approach are explained in next section.

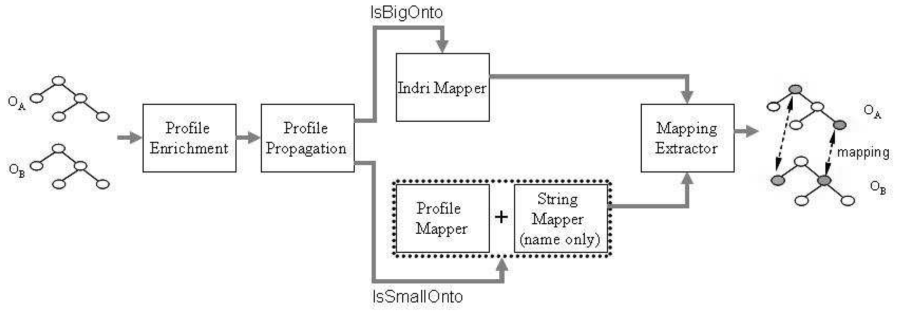


Figure 1 The architecture of PRIOR system

1.2 Specific techniques used

We introduce the term “profile”. Similar to the virtual document used in Falcon-AO system [11], the profile of a concept is a combination of all linguistic information of the concept, i.e. the profile of a concept = the concept’s name + label + comment + property restriction + other descriptive information. The Profile Enrichment is a process of using a profile to represent a concept in the ontology, and thus enrich its information. The purpose of profile enrichment is based on the observation that though a name is always used to represent a concept, sometimes the information carried in a name is restricted. While, other descriptive information such as comments may contain words that better convey the meaning of the concept.

The Profile Propagation exploits the neighboring information of each concept. That is, we pass the profile of the ancestors, children or siblings of the concept to the profile of the concept itself. The reason why we do profile propagation is based on the observation that if we see the taxonomic tree of an ontology as the index of a book, the super class in the ontology reflects the “context” of its subclasses and each subclass is the “content” of its super class. The process of profile propagation can be represented as: $V_{Nnew} = \sum_{N' \in S} w(N, N')V_{N'}$, where N and N’ represent two concepts in the ontologies, S represents the set of all concepts in the ontologies, V_{Nnew} represents the new profile vector of the concept N, $V_{N'}$ represents the profile vector of the concept N’, and w(N, N’) is a function that assigns different weights to the neighbors of the concept according to the distance between them. Two principles to assign the weight are applied: 1) The closer the two concepts are, the higher weight will be assigned, i.e. the weight of a parent is higher than the weight of a grandparents and the weight of a child is higher than the weight of a grandchild. 2) The weight of a parent is higher than the weight of a child and the weight of a child is higher than the weight of a sibling. This is because children inherit all characteristics of the parent and may extend some characteristics that parent doe not have, and sibling is usually a complementary of the concept.

For small ontologies, the Profile Mapper compares each concept of the ontologies by computing cosine similarity of the profile of each concept. Simultaneously, the String Mapper computes the similarity between the names of different concepts using Levenshtein distance. The profile similarity and the name string similarity are further integrated to obtain final similarities between concepts. However, if the ontology is too large, calculating the similarity matrix will require too many computing resources and it is time consuming. Based on the understanding that ontology mapping is also an information retrieval task, we turn to classic information retrieval method to solve the problem. Specifically, we integrated indri[1] search engine into PRIOR system. First, the Indri Mapper uses Indri to index profiles of concepts in ontology A. Then queries are generated based on the profiles of the concepts in ontology B. After storing the top-ranked results returned by the queries, we switch two ontologies, i.e. this time ontology B is indexed and queries are generated based on ontology A. The Indri Mapper will pass two sets of search results to the Mapping Extractor.

Having the similarity matrix obtained from small ontologies or Indri search results from large ontologies, the Mapping Extractor extracts all candidates of matched concepts and output the results in desired format.

1.3 Adaptations made for the evaluation

We didn’t do any major adaptations in order to align the OAEI campaign ontologies. However, for food test, we treat <skos:broader> and <skos:narrower> as parent and child relations.

1.4 Link to the system and parameters file

The system is available at: http://www.sis.pitt.edu/~mingmao/om06/

1.5 Link to the set of provided alignments (in align format)

The result file can be downloaded from http://www.sis.pitt.edu/~mingmao/om06/result.zip

2 Results

In this section we present the results of alignment experiments on OAEI 2006 campaign. All tests are run on a stand-alone PC running Fedora 4 operating system.

---

[1] http://www.lemurproject.org/indri

The PC has Pentium 4, 3.0GHz processor, 1G memory, 100GB Serial ATA hard disk and SUN JAVA VM 1.5.0_06.

2.1 benchmark

The benchmark tests can be divided into two types. Test 101-266 are systematically generated from reference ontology, in which some information are discarded, and test 301-304 are real bibliographic ontologies. Since our approach is relied on the linguistic information, we obtain high precision and recall where the test ontologies contain the same names (or name conventions) and/or comments as the reference ontology (i.e. test 101, 103, 104, 203, 204, 208, 221-247). However our approach fails in the recall where both name and comments are replaced or missing in the test ontologies (i.e. test 202, 248-266). For tests 201, 206-207 and 210, though the class name has been “removed” or expressed in another language, we can find some matched classed and properties due to the information of comments and instances. For tests 205 and 209 having name synonyms, the performance of our approach is not good because we do not use thesaurus. For real ontologies 301-304, they cover the same domain as reference ontology using similar descriptive information and different structural information. The result of these real tests shows the average performance of our approach is around 80%. The full result of all tests can be found in Appendix.

2.2 directory

The directory real world case consists of aligning web sites directory. It has 4640 elementary tests. Each of them is represented by pairs of OWL ontologies, where classification relation is modeled as OWL subClassOf. Therefore all OWL ontologies are taxonomies, i.e., they contain only classes (without Object and Data properties) connected with subclass relation. We use the same set of parameters and approach as those of benchmark test to obtain alignment results.

2.3 anatomy

The anatomy task is to find alignment between classes in two medical ontologies, FMA ontology and OpenGALEN ontology. FMA has 72559 classes and OpenGALEN has 9564 classes. Due to the huge size of the ontologies, we use Indri approach. Finally 2583 pairs of candidates have been found within 9 minutes.

2.3 food

The food thesaurus mapping task requires to create alignment between the SKOS version of the United Nations Food and Agriculture Organization (FAO) AGROVOC thesaurus, which has around 16000 terms and is expressed in multilingual, and the United States National Agricultural Library (NAL) Agricultural thesaurus, which has around 41000 terms and is expressed in monolingual. AGROVOC has 28179 concepts, and NAL has 41594 concepts. Due to the similar reason as anatomy task that the size of food thesaurus is too large, we use Indri approach. Finally 11511 pairs of candidates have been found within 73 minutes. Although “narrowMatch” and “broadMatch” are allowed, we can only get “exactMatch”.

3 General comments

3.1 Comments on the results

Since our approach relies on linguistic information such as name, label, comment, and other descriptive information, it can not handle pure graph matching task, like test 248-266 in benchmarks. Also we do not use external resources like WorldNet to process synonyms, which we believe is important in real cases. Furthermore, some ontology like AGROVOC contains labels in foreign languages; currently we do not use this type of information.

We use Alignment API to parse ontologies and generate alignments. When processing FMA ontology in anatomy test, the API reads each owl:Class as a class first and then as an individual one more time. In all properties of a class, only “ID” and “label” are assigned to the class, all other properties such as “part” and “constitutional_part” are assigned to the individual. Since only classes are alignment candidates, we miss all information in individual.

3.2 Discussions on the way to improve the proposed system

One possible improvement is to integrate external resources to increase recall. For instance, WordNet can be integrated to process synonyms and dictionaries can be used to process foreign languages. Another possible improvement is to find out a better way to adjust the propagation weights. It’s possible to train the weights with some training data.

3.3 Comments on the OAEI 2006 test cases

The ontologies in anatomy and food tests are very large and in a different format (i.e. SKOS, Protégé exported RDF) other than benchmark tests. It will be better to have a small part of ontology as training ontology, for which alignments are provided to participants. So that participants can train their approach on this training ontology. We also would like to see the OAEI 2006 campaign to be the first one to provide reference alignment for real word large scale ontologies so that different approaches can be judged in systematic way.

3.4 Comments on the OAEI 2006 measures

Considering the mapping relations in food track, the evaluation process is more complex. If concept A is an “exactMatch” to concept B, and concept C is a “broader” concept of B, then we can say concept A and C has a “broadMatch” relation. First we don’t know whether A-exactMatch-B and A-broadMatch-C will both appear in reference alignment. Second, if they both appear in reference alignment, but only A-exactMatch-B mapping is in an answer alignment, how do we calculate recall regarding A-broadMatch-C mapping?

4 Conclusion

In this paper, we briefly present a system for ontology mapping – PRIOR system, in which we explore linguistic and structural information and profile propagation method to process small ontologies. We also integrate classic information retrieval method to process large ontologies. The preliminary results are carefully analyzed and some future work are discussed.

References

1. Dhamankar, R., Y. Lee, et al. (2004). "iMAP: Discovering Complex Semantic Matches between Database Schemas." Proceedings of the International Conference on Management of Data (SIGMOD).
2. Doan, A., P. Domingos, et al. (2001). Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. SIGMOD Conference.
3. Embley, D. W., D. Jackman, et al. (2001). Multifaceted Exploitation of Metadata for Attribute Match Discovery in Information Integration. Workshop on Information Integration on the Web.
4. Gruninger, M. and J. Kopena (2003). "Semantic integration through invariants." Workshop on Semantic Integration at ISWC.
5. Hovy, E. (1998). Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In The First International Conference on Language Resources and Evaluation (LREC), Granada, Spain.
6. Jaro, M. A. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association 84:414–420.
7. Li, W., C. Clifton, et al. (2000). "Database integration using neural network: implementation and experience." Knowledge and Information Systems 2(1): 73-96.
8. Madhavan, J., P. Bernstein, et al. (2001). Generic Schema Matching Using Cupid. Proceedings of the 27th International Conference on Very Large Data Bases (VLDB), Roma (IT), Morgan Kaufmann Publishers Inc.
9. Melnik, S., H. Garcia-Molina, et al. (2002). Similarity flooding: a versatile graph matching algorithm and its application to schema matching. Proc. 18th International Conference on Data Engineering (ICDE), San Jose (CA US).
10. Mitra, P., G. Wiederhold, et al. (1999). Semi-automatic integration of knowledge sources. Proc. of the 2nd Int. Conf. On Information FUSION'99.

11. Qu, Y., Hu, W., and Cheng, G. 2006. Constructing virtual documents for ontology matching. In Proceedings of the 15th International Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 23-31.
12. Rahm, E. and P. Bernstein (2001). "A survey of approaches to automatic schema matching." The VLDB Journal 10(4): 334-350.

Appendix: Raw results

Matrix of results

#	Name	Prec.	Rec.
101	Reference alignment	1	1
102	Irrelevant ontology	0.00	NaN
103	Language generalization	1	1
104	Language restriction	1	1
201	No names	0.94	0.32
202	No names, no comments	0.6	0.03
203	No comments (was misspelling)	1	1
204	Naming conventions	1	0.94
205	Synonyms	0.63	0.42
206	Translation	0.96	0.7
207		0.96	0.7
208		1	0.93
209		0.53	0.3
210		0.94	0.53
221	No specialisation	1	1
222	Flatenned hierarchy	1	1
223	Expanded hierarchy	1	1
224	No instance	1	1
225	No restrictions	1	1
228	No properties	1	1
230	Flattened classes	0.94	1
231*	Expanded classes	1	1
232		1	1
233		1	1
236		1	1
237		1	1
238		1	1
239		0.97	1
240		0.97	1
241		1	1
246		0.97	1

247		0.97	1
248		0.33	0.01
249		0.6	0.03
250	Individual is empty	1	0.06
251		0.6	0.03
252		0.5	0.03
253		0.33	0.01
254		NaN	0
257		1	0.06
258		0.6	0.03
259		0.5	0.03
260		0.5	0.03
261		0.5	0.03
262		NaN	0
265		0.5	0.03
266		0.5	0.03
301	Real: BibTeX/MIT	0.92	0.74
302	Real: BibTeX/UMBC	0.86	0.63
303	Real: Karlsruhe	0.68	0.82
304	Real: INRIA	0.95	0.96

Result of Ontology Alignment with RiMOM at OAEI'06

Yi Li, Juanzi Li, Duo Zhang, and Jie Tang
Department of Computer Science and Technology, Tsinghua University
{ly, ljz, zhangduo, tangjie}@keg.cs.tsinghua.edu.cn

Abstract. In this report, we briefly describe our system RiMOM and its underlying techniques. Given two ontologies, RiMOM intends to combine multiple strategies, aiming at finding the "optimal" alignments from the source ontology to the target one. RiMOM integrates multiple strategies: edit-distance based strategy, statistical-learning based strategy, and three similarity-propagation based strategies. Each strategy is defined based on one kind of ontological-information/approach. RiMOM conducts alignment finding as follows. It first estimates two factors respectively approximately representing the structure similarity and the label similarity of the two ontologies. The two factors are used in strategy selection to determine which strategies will be used in the alignment task. Then, we apply the selected strategies to find the alignment independently and combine the alignment results. Finally we employ the alignment refinement to prune "unbelievable" alignments. This report presents our results based on the evaluation. We also share our thoughts on the experiment design, showing specific strengths and weaknesses of our approach.

1. PRESENTATION OF THE SYSTEM

Ontology alignment is the key point to reach interoperability over ontologies. In semantic web environment, ontologies are usually distributed and heterogeneous and it is necessary to find the mapping between them before processing across them. In recent years, much research work has been conducted for finding the alignment of ontologies [1] [4].

RiMOM is a tool for ontology alignment by combining different strategies, aiming at finding the "optimal" alignment results [5]. Each strategy is defined based on one kind of information or one type of approach. In our current version, there are five strategies defined: edit-distance based strategy, statistical-learning based strategy, and three similarity-propagation based strategies (including concept-to-concept propagation strategy (CCP), property-to-property propagation strategy (PPP), and concept-to-property propagation strategy (CPP)).

1.1 State, purpose, general statement

We here define ontology alignment as a directional one. Given an alignment from ontology O_1 to O_2, we call ontology O_1 as source ontology and O_2 as target ontology. We call the process of finding the alignment from O_1 to O_2 as (Ontology) alignment discovery or alignment finding.

Challenges for automating ontology alignment include: 1) how to automatically find alignments of high quality; 2) how to find the alignments efficiently; 3) how to make full use of the user interaction, since entirely automatic alignment is usually not possible; 4) how to automatically adjust the strategies for finding the alignments in a specific task, since the characteristics of the ontologies to be aligned are different in different tasks; 5) how to ease parameterizing, as the accuracy of alignments may vary largely with different parameters.

In this campaign, we focus on dealing with the problems of 1), 2), and 4) with our system RiMOM.

1.2 Specific techniques used

There are six major steps in the alignment process of RiMOM:

1) Similarity factors estimation. Given two ontologies, it estimates two similarity factors, which respectively approximately represent the structure similarity and the label similarity of the two ontologies. The two factors are used in the next step of strategy selection.

2) Strategy selection. The basic idea of strategy selection is if two ontologies have high label similarity factor, then RiMOM will rely more on linguistic based strategies; while if the two ontologies have high structure similarity factor, then we will employ similarity-propagation based strategies on them. See Section 1.2.2 for details.

3) Strategy execution. We employ the selected strategies to find the alignment independently. Each strategy outputs an alignment result.

4) Alignment combination. It combines the alignment results obtained by the selected strategies. The combination is conducted by a linear-interpolation method.

5) Similarity propagation. If the two ontologies have high structure similarity factor, RiMOM employs an algorithm called similarity propagation to refine the found alignments and to find new alignments that can not be found using the other strategies. Similarity propagation makes use of structure information.

6) Alignment refinement. It refines the alignment results from the previous steps. We defined several heuristic rules to remove the “unbelievable” alignments.

1.2.1 Multiple strategies

The strategies defined in RiMOM can be classified into two categories: linguistic based strategies and structure based strategies.

1. Linguistic based strategies

RiMOM contains two kinds of linguistic based strategies: edit-distance based strategy and statistical-learning based strategy. In our current version of RiMOM, for the statistical-learning based strategy, we use the classification method of K-Nearest Neighbor (KNN). For facilitating the description, we hereafter write the two strategies as ED and KNN.

In ED, we calculate the edit distance between labels of two entities. In KNN, we formalize the problem of alignment as a problem of text classification. We view $e_2 \in O_2$ as a class and its label, comment, and instances as a ‘document’ of the class. The text in a ‘document’ is tokenized into words. Then we employ stemming and stop words removing on the words and view the remains as features to train a text classification model. We also add some other general features which prove to be very helpful. For a concept, the features include: the number of its sub concepts, the number of properties it has, and the depth of the concept from “OWL:Thing”.

For finding the alignment, we use the same method to generate a ‘document’ for a concept $e_1 \in O_1$ and also add the general features as that in building the classification model. Then we use the trained classification model to identify which class the document should be classified. In this way, we are able to find which entity in O_2 is the most possible one for an entity $e_1 \in O_1$ to be aligned.

The two strategies can be used for finding alignments independently. They can also be used together. In the latter case, we combine alignments of different strategies by:

$$Map(e_1, e_2) = \frac{\sum_{k=1\ldots n} w_k \sigma\left(Map_k(e_1, e_2)\right)}{\sum_{k=1\ldots n} w_k} \quad \textbf{(1)}$$

where $e_1 \in O_1$ and $e_2 \in O_2$; $Map_k(e_1,e_2)$ is the alignment score obtained by strategy k. w_k is the weight of strategy k. σ is a sigmoid function, which is defined as $\sigma(x) = 1/\left(1+e^{-5(x-\alpha)}\right)$, where α is tentatively set as 0.5.

2. Structure based strategies

The structure information in ontologies is useful for finding the alignments especially when two ontologies share the common/similar structure. According to the propagation theory [2], we define three structure based strategies in RiMOM, namely concept-to-concept propagation strategy (CCP), property-to-property propagation strategy (PPP), and concept-to-property propagation strategy (CPP).

Intuition of the propagation based method is that if two entities are aligned, their super-concepts may also be aligned. The basic idea of the method is to propagate the similarity of two entities to entity pairs with some kinds of relationship with them, for example, subClassOf, superClassOf, siblingClassOf, subPropertyOf, superPropertyOf, range, and domain (superClassOf is not defined in OWL, it is viewed as the converse relationship of subClassOf. Likewise for superPropertyOf. siblingClassOf is not defined also in OWL. It means that the two concepts have the same super concept). The idea is inspired by the algorithm of similarity flooding proposed for schema matching [3]. We extended the algorithm and adaptively used them in the three structure based strategies. Details of the method will be reported elsewhere.

In CCP, we propagate similarities of concepts pair across the concept hierarchical structure. In PPP, we propagate similarities of property pair across the property hierarchy. In CPP, we propagate similarities of concepts pair to their corresponding property pair, and vice versa.

The structure based strategies are employed after the linguistic based strategies. They can be used to adjust the alignments and find new alignments.

1.2.2 Similarity factors estimation

Our preliminary experiments show that the multi-strategy based alignment does not always outperform its single-strategy counterpart. We then consider three questions: (1) for a new, unseen mapping task, should we select a multi-strategy based solution or just one single-strategy based solution? (2) if the task is suitable to use multiple strategies, then which strategies should be selected so as to obtain better alignment results? (3) the method for strategy selection needs to be efficient. This is important because for aligning large-scale ontologies, the efficiency may be a critical problem. We propose to deal with the problems by using similarity factors estimation.

Given two ontologies: source ontology O_1 and target ontology O_2, we calculate two approximate similarity factors: structure similarity factor and label similarity factor.

We define structure similarity factor as:

$$F\_SS = \frac{\#common\_concept}{\max(\#nonleaf\_c_1, \#nonleaf\_c_2)} \tag{2}$$

where $\#nonleaf\_c_1$ indicates the number of concepts in O_1 that has sub concepts. Likewise for $\#nonleaf\_c_2$. $\#common\_concept$ is calculated as follows: if concepts $c_1 \in O_1$ and $c_2 \in O_2$ have the same number of sub concepts and they are in the same depth from the concept "owl:Thing", we add one to $\#common\_concept$. After enumerated all pair, we obtain the final score of $\#common\_concept$. Intuition of the factor is that the larger the structure similarity factor, the more similar the structures of the two ontologies are.

The label similarity factor is defined as:

$$F\_LS = \frac{\#same\_label}{\max(\#c_1, \#c_2)} \tag{3}$$

where $\#c_1$ and $\#c_2$ respectively represent the number of concepts in O_1 and O_2. $\#same\_label$ represents the number of pairs of concepts $\{(c_1, c_2) | c_1 \in O_1 \text{ and } c_2 \in O_2\}$ that have the same label.

The two factors are defined simply and not used to accurately represent the real "similarities" of structures and labels. However, they can approximately indicate the characteristics of the two ontologies. Moreover, they can be calculated efficiently.

So far, we carried out the strategy selection by heuristic rules. For example, if the structure similarity factor F_SS is lower than 0.25, then RiMOM suppresses the CCP and PPP strategies. However, the CPP will always be used in the alignment process.

1.3 Adaptations made for the evaluation

No special adaptations have been made. However, some parameters are tuned and set in the experiments. For example, for strategies combination (cf. equation 1), we set the weight of ED as 0.5 and that of KNN as 1. For strategy selection, we define 0.25 as the threshold to determine whether CCP and PPP will be suppressed or not. We also define 0.2 as threshold to determine whether ED will be suppressed or not.

1.4 Link to the system, parameters file, and provided alignments

Our system RiMOM (including the parameters file) can be found at http://keg.cs.tsinghua.edu.cn/project/RiMOM/. For details of the approach, see [5].

The alignment results of the campaign are available at http://keg.cs.tsinghua.edu.cn/project/RiMOM/OAEI2006/.

2 Results

RiMOM has been implemented in Java. We use OWL-API to parse the RDF and OWL files. The experiments were carried out on a Server running Windows 2003 with two Dual-Core Intel Xeon processors (2.8 GHz) and 3-gigabyte memory. All the alignments outputted by RiMOM are based on the same parameters.

2.1 Benchmark

2.1.1 Tests 101-104

The tests 101, 103, and 104 are basic tests for ontology alignment. The source ontologies contain concepts and properties with the same names as those in the reference ontologies.

Both linguistic based strategies and structure based strategies were employed for finding the alignment (because both label similarity and structure similarity factors exceed the thresholds), however, as linguistic based strategies can easily find most of the alignments, the structure based strategies took little effect to the final results. In test 102, RiMOM outputs no alignment.

In the three tests (excluding test 102), both precision and recall are 1.0. The average time cost is 3.36s.

2.1.2 Tests 201-210

Tests 201 through 210 have high structure similarity factor (equal to 1.0) with the reference ontology. Some of the tests have high label similarity factor (e.g. test 203), some have synonym labels with the reference ontology (e.g. test 205 and 209), and some others have low label similarity factor (e.g. tests 201, 202, 206, 207, and 210).

Using the strategies selection method, we are able to apply different strategies in the different tests. For example, for test 201, where label of concepts and properties are replaced by a random ones, ED is suppressed and KNN and the structure based strategies are active. Using KNN, we can find some matched concept pairs and property pairs. Then based on the matched pairs, we utilize the structure based strategies to find the other alignments that cannot be found by KNN.

In the ten tests, precision ranges from 0.88 to 1.0 and recall stays between 0.82 and 1.0. The average time cost is 2.638s.

2.1.3 Tests 221~247

For most of these tests the structures are changed, which means that the structure similarity factors are low, however the label similarity factors are very high.

For tests that have low structure similarity factors, we suppress the structured based strategies, for example, tests 221, 232, 233, and 241. (Note: CPP is still active.) For tests that have both high label similarity factor and structure similarity factor, both linguistic based strategies and structure based strategies were employed, although structure based strategies made little contribution.

In these tests, precision ranges from 0.94 to 1.0 and recall equals to 1.0. The average time cost is 1.99s.

2.1.4 Tests 248~266

These tests were the most challenging ones to our approach. Labels and comments have been removed and structures have also been changed as well. In this case, both label similarity factor and structure similarity factor between the source ontologies and the reference ontology are low. For most of the tests, we found that KNN is the most useful one and the other strategies take little effects. In tests 249, 250, and 257, the structure based strategies took effect to help improve the final alignments.

In these tests, precision ranges from 0.73 to 1.0 and recall stays between 0.27 and 0.82. The average time cost is 1.59s.

2.1.5 Tests 301~304

In tests 301-304, the source ontologies are from real world, modeled by different institutions but for the same domain of bibliographic metadata. The real-world tests combine the difficulties of the previous tests.

In the tests, based on the strategy selection method, both linguistic based strategies and structure based strategies were employed except the test 301, where we only applied linguistic based strategies.

In these tests, precision ranges from 0.77 to 0.9 and recall stays between 0.69 and 0.97. The average time cost is 3.14s.

2.2 directory

The directory ontologies are organized as a taxonomy with sub-sumption hierarchies. We use two methods to obtain the alignment results. The first one was obtained by using RiMOM with the same set of parameters as the ones for benchmark test. Both linguistic based strategies and structure based strategies were employed in this task. The results seem to be not ideal.

The other alignment result was obtained by a specific version of RiMOM, called RiMOM-directory. In RiMOM-directory, except ED and KNN, we also integrate another strategy based on Wordnet, one of the most popular thesauruses (called as Wordnet hereafter). Because in directory alignment, there are many synonym words used in the labels, Wordnet is expected to be useful. We also made some other adaptation, for example, for structure based strategies we only use CCP, as there is no property and instances in the directory data (also in CCP, we only consider the relationship “OWL:subClassOf”).

We obtained three alignment results using RiMOM-directory with different strategies: 1) linguistic based strategies (including ED, KNN, and Wordnet) only. In this case, the precision, recall, and F1-measure are 0.36, 0.33, and 0.35 respectively; 2) both linguistic based strategies and the CCP strategy (with only one iteration of propagation). The precision, recall, and F1-measure are 0.39, 0.40, and 0.40 respectively; 3) same setting as that in 2) but with *n* iterations. The precision, recall, and F1-measure are 0.38, 0.40, and 0.39 respectively.

2.3 anatomy

RiMOM met problems in parsing the anatomy ontologies and finally outputs no alignments.

2.4 food

The ontologies in the food test are large and RiMOM suppressed the structure based strategies and applied only a simple version of the linguistic based strategies for finding the alignment.

3 General comments

3.1 Comments on the results

An objective and comprehensive comment on strengths or weakness requires the comparison with other participants, which are not available so far (will be available before the workshop). Here, we share some thoughts about the results.

Strengths

From experimental results, we see that RiMOM can achieve high performance when the ontologies to be aligned have similar linguistic information or similar structure information. Some concluding remarks are summarized as follows:

1) Linguistic information (including label of concepts and properties) is important and help to align most of the entities.

2) Structure information can be used to improve the alignments, in particular when linguistic information is missing.

3) Strategy selection is important. In different alignment tasks, the ontologies to be aligned have different characteristics, it would be particularly helpful to find the characteristics of the ontologies and apply correspondingly strategies on them.

4) Alignment refinement is helpful. In refinement, we removed the unbelievable alignments, which improves the precision in many tests.

5) RiMOM can find the alignment quickly. The time costs range from 0.69s to 6.70s in the benchmark tests.

Weakness

1) RiMOM cannot deal with large-scale ontologies. The biggest problem here is that our structure base strategies cannot efficiently do the propagation in the large graph (by viewing the ontology as a graph).

2) We met problems when dealing with the anatomy ontologies.

3) We note that parameter setting is very important. We have found that using different parameter settings, with the exactly same approach, the alignment results may differ largely. So far, we tuned the parameters manually. It is not adaptable in particular when the ontologies are very large, which means that tuning different parameters to find the best ones is not possible.

3.2 Discussions on the way to improve the proposed system

Possible improvements are corresponded to the related weaknesses in the previous section.

1) Our proposal is to partition the large ontologies into small slices and then employ the structure based strategies on the slices.

2) Efforts are being made to integrate a more powerful parser into our system.

3) Our thinking is to use a supervised machine learning method to find the optimal parameters based on some training data sets.

3.3 Comments on the OAEI 2006 test cases

The benchmark tests indicate very interesting general results on how the alignment approach behaves. These tests are really useful, as a good underlying test base, for evaluating and improving the alignment algorithm and system.

For future work, it might be interesting to add some tests to evaluate the time cost of systems, as for large-scale ontology alignment, the issue of efficiency may be critical.

4 Conclusion

In this report, we have briefly introduced our approach and the tool, that is called RiMOM, for finding ontology alignment. We have presented the alignment process of RiMOM and explained each step. We applied the tool to the test data and the experimental results show that our proposed approach can achieve high performance quickly. We summarized the strengths and the weaknesses of our proposed approach and gave possible improvement for the system in the future work.

References

[1] J. Euzenat. State of the art on ontology alignment. http://www.inrialpes.fr/exmo/cooperation/kweb/heterogeneity/deli/. August, 2004.

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. International Journal of Computer Vision, Vol. 70, No. 1, October 2006.

[3] S. Melnik, H. Garcia-Molina and E. Rahm: Similarity Flooding: a versatile graph matching algorithm and its application to schema matching. In Proc. of 18th ICDE. San Jose CA, Feb 2002. pp. 117-128

[4] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. The VLDB Journal, 2001, 10:334-350.

[5] J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang. Using Bayesian decision for ontology mapping. Journal of Web Semantics. To be published.

Appendix: Raw results

The following results were obtained in the evaluation runs.

Matrix of results

#	Name	Prec.	Rec.	Time
101	Reference alignment	1.00	1.00	4.72s
102	Irrelevat ontology	N/A	N/A	
103	Language generalization	1.00	1.00	2.84s
104	Language restriction	1.00	1.00	2.52s
201	No names	1.00	1.00	1.98s
202	No names, no comments	1.00	0.82	1.56s
203	No comments	1.00	1.00	2.53s
204	Naming conventions	1.00	1.00	2.72s
205	Synonyms	1.00	0.99	3.88s
206	Translation	1.00	0.99	5.16s
207		1.00	0.99	2.27s
208		0.98	0.98	2.48s
209		0.88	0.87	1.88s
210		0.99	0.89	1.92s
221	No specialisation	1.00	1.00	2.28s
222	Flatenned hierachy	1.00	1.00	2.58s
223	Expanded hierarchy	1.00	1.00	4.83s
224	No instance	1.00	1.00	2.75s
225	No restrictions	1.00	1.00	2.66s
228	No properties	1.00	1.00	1.00s
230	Flatenned classes	0.94	1.00	2.19s
231		1.00	1.00	3.00s
232		1.00	1.00	2.13s
233		1.00	1.00	0.77s
236		1.00	1.00	0.94s
237		0.99	1.00	2.42s
238		1.00	1.00	3.52s
239		0.97	1.00	1.08s
240		0.97	1.00	1.06s
241		1.00	1.00	0.69s
246		0.97	1.00	1.06s
247		0.97	1.00	1.02s
248		1.00	0.81	2.03s
249		1.00	0.82	1.74s
250		1.00	0.55	0.92s
251		0.74	0.59	1.80s
252		0.84	0.71	4.80s
253		1.00	0.81	1.61s
254		1.00	0.27	0.72s
257		1.00	0.55	0.89s
258		0.73	0.59	2.19s
259		0.84	0.71	2.17s
260		0.87	0.45	1.03s

261		0.82	0.27	1.33s
262		1.00	0.27	0.69s
265		0.87	0.45	1.03s
266		0.82	0.27	0.92s
301	BibTeX/MIT	0.82	0.74	1.78s
302	BibTeX/UMBC	0.77	0.69	1.73s
303	Karlsruhe	0.77	0.84	6.70s
304	INRIA	0.90	0.97	2.36s

When usual structural alignment techniques don't apply

Chantal Reynaud[1] and Brigitte Safar[1]

University of Paris Sud-CNRS (LRI), INRIA (Futurs),
LRI, Building 490, 91405 Orsay Cedex, France
{chantal.reynaud, brigitte.safar}@lri.fr
http://www.lri.fr/~cr

Abstract. This paper deals with taxonomy alignment. It presents structural techniques of an alignment method suitable with a dissymmetry in the structure of the mapped taxonomies. The aim is to allow a uniform access to documents belonging to a same application domain, assuming retrieval of documents is supported by taxonomies.

1 Introduction

Our work focuses on taxonomy alignment techniques. Indeed, we assume that the description of the content of most todays information systems is often not very much specified and is based on very simple ontologies reduced to classification structures, i.e. taxonomies. Moreover, we suppose that the structures of the taxonomies that have to be aligned are heterogeneous and dissymmetric, one taxonomy being deep whereas the other one is flat. In this context, the approaches which relied on OWL data representations exploiting all the ontology language features don't apply. Similarity of two entities cannot be identified based on their similar properties or on the status of their respective parents and siblings, because these data are not available. We can only use the following available data: labels of concepts in both taxonomies, the structure of the deeper taxonomy and external linguistic resources such as WordNet.

The contribution of this paper is a mapping process composed of a sequence of various techniques designed to make best use of the characteristics of the taxonomies: very specialized taxonomies with only sub-class links, concepts with labels which are expressions composed of a lot of words, words common to a lot of labels. We classify the found mappings into two groups according to their relevance: probable mappings and potential mappings to be confirmed. The mapping process is generic, usable across application areas. It has been evaluated on real-world taxonomies and on test taxonomies extracted from a repository about ontology matching [6]. Experiments showed that the proposed techniques give very relevant mappings when the aligned taxonomies have the same characteristics as the taxonomies having motivated our approach.

2 The alignment approach

The objective of our approach is to generate mappings between taxonomies. For us, a taxonomy is a pair (C, H_C) consisting of a set of concepts C arranged in a subsumption hierarchy H_C. A concept is only defined by two elements: a label and subclass relations. The label is a string which can be an expression composed of several words. Subclass relations establish links with other concepts. It is the single semantic association used in the hierarchy. A taxonomy is generally represented by an acyclic graph where concepts are nodes connected by directed edges corresponding to subclass links. Given two structurally dissymmetric taxonomies, the objective is to map the concepts of the less structured one, the source taxonomy T_S, with concepts of the more structured one, the target taxonomy T_T. The alignment process is oriented from T_S to T_T. The goal is to find one-to-one mappings. Relations can be of two kinds: equivalence (*isEq*) and subclass (*isA*). So, for each concept c_S in T_S, we try to find a corresponding concept, c_T in T_T, linked to c_S with an equivalence or a subclass relation.

Alignment is based on the Lin similarity measure [1] computed between each concept c_S in T_S and all the concepts in T_T. This measure compares the tri-grams of the labels and has been adapted to take into account the importance of words inside expressions. From the measurements we compute MC, the set of mapping candidates of a concept c_S in T_S. MC includes concepts of T_T which have a high similarity value with c_S (only the three most similar concepts b_1, b_2, b_3 are retained) and Inc, the set of concepts of T_T with a label included in the label of c_S. Various techniques (terminological and structural cf.Fig.1) are then applied in sequence to select the most relevant concept among all the mapping candidates [3]. We are going to show that the most relevant concept is not necessarily the one with the highest similarity measure. Terminological techniques are executed first. In default of place, they will not be detailed here. These techniques lead to mappings which are generally reliable but not always sufficiently numerous. Therefore, they are completed by the structural mappings described in the next section. These latter techniques define a mapping as a correspondence between close concepts. If the suggested mapping from c_S to c_T is wrong, then the right mapping will be a relation from c_S to c'_T, with c'_T close to c_T in the taxonomy. It is a guide for the user who will not have to browse the whole target taxonomy when studying the results of the system.

```
$TaxoMap(T_S, T_T)$
For each $c_S \in T_S$ do
    For each $c_T \in T_T$ do $Sim_{LinLike}(c_S, c_T)$
    $MC \leftarrow$ MappingCandidates($c_S$)
    If TerminologicalTechniques($c_S, MC$) then stop
    Else StructuralTechniques($c_S, MC$)
```

Fig. 1. The Alignment process

3 Exploiting structural features

The two techniques presented in this section are structure based techniques leading to the discovery of subclass mappings. The first technique is performed on T_T whose structure is supposed to be the deepest. Then we use $WordNet$ [2], exploiting its structure and its semantic relations.

3.1 Exploiting the structure of the target taxonomy: STR_T

This technique, denoted STR_T, works on MC, the set of mapping candidates of a concept c_S in T_S. The idea is to exploit the location of the elements of MC in T_T. Their proximity in the graph is considered to be a semantic proximity. We therefore try to identify the sub-graph rooted in a node associated to a concept which is not too general and such that this sub-graph groups a maximum of nodes of MC. It will represent a relevant context shared by most of the mapping candidates. We then consider that the involved concept c_S may be mapped with a node of this sub-graph. STR_T relies on the computation of the Lowest Common Ancestor, LCA, of a set of nodes in a graph, which is the node of greatest depth which is an ancestor of all the nodes of the set. Our goal is to find a LCA of the elements of MC which is not too high in the taxonomy. However the LCA node of a set of elements is all quite high in the graph since the elements are very distant from each other. We propose a measure, the relative density (DR), to evaluate sub-graphs grouping nodes of a sub-set of MC. For each sub-graph rooted in Anc, the LCA node, and grouping MC_{Anc} nodes, we compute $DR(Anc)$.

$DR(Anc)$ relies on three criteria: (1) the number of elements in MC_{Anc}, (2) Sim_{Lin_Like}, the similarity between the elements in MC_{Anc} and c_S,(3) the distance as the number of edges on the paths from each element of MC_{Anc} to Anc.

$$DR(Anc) = \frac{|MC_{Anc}| * \sum_{C_T \in MC_{Anc}} Sim_{LIN\text{-}Like}(c_S, c_T)}{|MC| * \sum_{C_T \in MC_{Anc}} dist(c_T, Anc)}$$

The sub-graph rooted in the Anc with the highest DR is considered to be the most relevant. C_{MaxAnc}, the node of this sub-graph with the highest similarity measure, will be the candidate selected for the mapping. If it belongs to Inc, the set of concepts with a label included into the label of c_S, it is suggested as a possible parent of the involved concept c_S. Otherwise, C_{MaxAnc} is proposed as a possible sibling and its parent (not necessarily Anc) will be suggested as a possible parent of c_S. As an example, Fig. 2 represents the sub-graph of T_T grouping the elements of $MC = \{b_1, b_2, b_3\} \cup Inc = \{$*beef*$\}$ for $c_S =$ *beef adipose tissue*. The node *Fresh meat* is the LCA for all the elements of MC with a distance of 7. However, *beef* is the LCA of three mapping candidates $\{$*beef*, b_2, $b_3\}$ with a distance of only 2. $DR($*beef*$)$ is the highest (cf.Fig.2). *beef connective tissue* is the node of this sub-graph with the highest similarity value to c_S. So *beef adipose tissue* will be a sibling of *beef connective tissue* and linked to *beef* with a

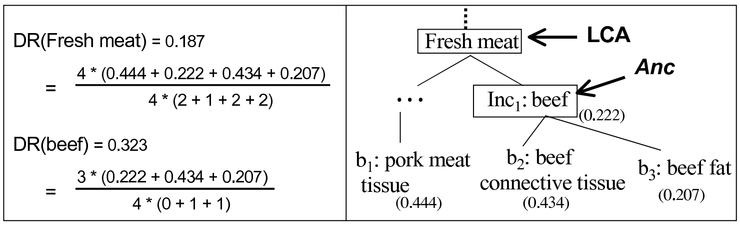


Fig. 2. Common ancestors and relative density

subclass relation. Note that this technique avoids mappings with concepts with a little higher similarity measure but meaningful in a context different from the one common to most of the MC (as b_1 in the example).

3.2 Exploiting the structure of *WordNet*: STR_W

The techniques seen up till now are not enough if the concepts are similar semantically but not syntactically. So, at that point, we propose to run STR_W. STR_W relies on the *hyperonymy/hyponymy WordNet* structure to find the concept of T_T semantically similar to each concept of T_S not yet mapped. STR_W will be able to map, for example, *cantaloupe* with *watermelon* which are not synonyms but two specializations of *melon.*

Running STR_W assumes that the application root node, denoted $root_A$, has already been identified. It is the most specialized concept in *WordNet* which generalizes all the concepts contained in the involved application domain. STR_W searches WordNet for the hypernyms of each term of T_S not yet mapped and of each term of T_T (according to all their senses) until $root_A$ is reached. For example, the result of a search on *cantaloupe* is two sets of hypernyms corresponding to two different senses.
Sense 1: *cantaloupe→sweet melon→melon→gourd →plant→ organism→Living thing*
Sense 2: *cantaloupe→sweet melon→melon→edible fruit→green goods→ food*

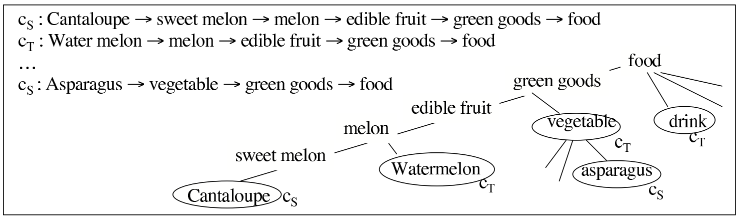


Fig. 3. A sub-graph of T_{wn} where *cantaloupe* and *watermelon* are related

Only the paths from the invoked terms to $root_A$ will be selected because they represent the only senses which are accurate for the application (sense 2 in the example, the application root node being *food*). So a sub-tree, denoted T_{wn}, is obtained. It is composed of all the terms and the relations of the retained paths (cf.Fig.3). For each concept c_S, STR_W selects in T_{wn} the most similar concept belonging to T_T using *Wu&Palmer*'s measure [5].

$$Sim_{W\&P}(c_1, c_2) = \frac{2 * depth\,(LCA(c_1, c_2))}{depth(c_1) + depth\,(c_2)}$$

According to the $sim_{W\&P}$ measure, the concept that is the most similar to a node c_S is its parent. Moreover, we showed in [3] that the similarity is higher between c_S and any of its siblings or any of the descendants close to its siblings than between c_S and its grandparent, until a depth p that can be computed for each node c_S in function of its depth in the tree. In the same way, we can compute the depth p' from which the similarity of the great-grandparent must be considered, and so on. Using these properties, we proposed an efficient strategy in [3] which does not require the computation of many similarity measurements.

4 Experiments and Discussion

Two kinds of experiments have been performed. First, experiments have been made in the setting of the e.dot project[1], on two real-world taxonomies in the field of predictive microbiology. Second, we applied our techniques on test taxonomies [6]. The latter are not structurally dissymmetric and cover a large domain. The application conditions of the techniques are not achieved but our objective is to make these tests in order to sketch some ideas to do improvements and to widen the scope of our approach. These experiments have shown where our specific strengths and weaknesses are. Whatever taxonomy we aligned, our approach was able to retrieve almost all the equivalence mappings given with the taxonomies. Furthermore, its strong point is to propose as a bonus a lot of other mappings (subclass mappings). Some mappings have a high precision and are then certain (likely mappings generated by the terminological techniques). Other ones (potential mappings generated by the structural techniques) are less certain (low precision) and have to be validated. This confirms the order in the application of our techniques. Concerning the structural techniques, STR_T proved to be very useful and leads to relevant mappings when concepts have labels composed of a lot of words and when some words are common to many labels. On the opposite, STR_W is all the more appropriate since the application domain is small. The real-world taxonomies which have motivated our approach gather all these characteristics, unlike the others. Better results are then obtained.

5 Conclusion

We described two structural techniques to align structurally asymmetric taxonomies. These techniques are original because different from a search of structural similarity in models. They are executed to suggest additional mappings. These mappings are not certain but they can be a good complement, if human involvement is possible, as experiments showed. We will continue this work by adapting and extending our techniques according to the experiment results. Our first objective is to be able to align taxonomies relative to larger application domains.

References

1. D. Lin. An Information-Theoretic Definition of Similarity, In *Proc. of the Int. Conf. on Machine Learning (ICML)*, pp. 296-304, 1998.
2. Miller, G. A. WordNet: A lexical Database for English. Communications of the ACM. (1995) Vol. 38(11) 39-45
3. C. Reynaud, B. Safar. Structural Techniques for Alignment of Taxonomies: experiments and evaluation, In *TR 1453, LRI, Univ. of Paris-Sud*, June 2006.
4. P. Shvaiko, J. Euzenat. A Survey of Schema-based Matching Approaches, In *Journal on Data Semantics*, 2005.
5. Z. Wu and M. Palmer. Verb Semantics and Lexical Selection, In *Proc. of 32nd Meeting of the Ass. for Computational Linguistics*, 1994.
6. http://www.ontologymatching/evaluation.html

---

[1] E.dot is a research project funded by national network on software technology (RNTL), 2003-2005.

A Framework for Combining Ontology and Schema Matchers with Dempster-Shafer

Paolo Besana

School of Informatics
University of Edinburgh

Abstract Ontologies, at least in the form of taxonomies, have proved rather successful, and are employed in many fields, as far apart as biology and finance. Reaching an agreement over a single ontology has proved difficult, and to obtain actual interoperability it is necessary to map the different ontologies. Mapping one entity between a source ontology and one in a target ontology means to compare the first entity with all the entities in the second ontology: matchers analyse different aspects of the entities to identify the similarities. A single matcher can analyse only some aspects, and often has to rely on uncertain information. Therefore combining the outcomes of different matchers can yield better results.
In this paper I present a framework that uses Dempster-Shafer as a model for interpreting and combining results computed by the matchers.

1 Introduction

Ontologies have proved to be a powerful tool, and they have become common. For example, ontologies in the form of taxonomies are used by Google and Yahoo to categorise websites and by Amazon and eBay to classify their products.

However, the development and the acceptance of a common ontology has failed to occur, and consequently a number of different ontologies are used. To exploit the richness provided by the ontologies it is necessary to build bridges between them. The various attempts to reconcile ontologies can be divided into *merging*, *integrating* and *mapping*, with mapping laying at the basis.

This paper[1] presents a framework for ontology mapping that uses Dempster-Shafer to interpret and combine the results computed by different matchers.

2 Ontology mapping as decision making under uncertainty

A mapping algorithm receives two ontologies and returns the relations (*equivalence,* etc) between their concepts. Stated otherwise, the algorithm finds the subsets $\Phi_1, \ldots, \Phi_n$ of the Cartesian product $O_1 \times O_2$ that contain the relations between the items in O_1 and O_2. This is obtained calling a *matcher* function that verifies to what degree μ each pair $\left\langle t^i_{O_1}, t^j_{O_2} \right\rangle$ belongs to the subset Φ_{rel}:

$$matcher\ :\ t \times t \times \Phi_{rel} \rightarrow \mu \qquad (1)$$

---

[1] full version available at: http://pyontomap.sourceforge.net

Mapping algorithms use different methods to identify relations between terms in the different ontologies. These methods assume that ontologies share similarities that can be found. For example, the similarities can be in the labels of the entities, in their formal definition, or in their description.

A general method for finding a mapping between a given entity $t \in O_{source}$ and an unknown entity $t_j \in O_{target}$ is to compare the given t with all the entities in O_{target}, and to keep the pair that belongs to the most significant relation (for example *equivalence*), with the highest membership degree μ:

$$mapper : t \times O_{target} \to \langle t_j, rel, \mu \rangle \tag{2}$$

More sophisticated methods can verify the consistency of the choice, and keep the strongest mapping that does not conflict with other mappings.
Different approaches of ontology mapping in literature can be classified by:

- ▷ *the binary relations they search*: some look only for similarity [3], other look for more complex ontological relations [6].
- ▷ *the methods they use for taking the decision*: some use only string comparison, others use thesauruses, others analyse the similarities in the structure of the ontologies [9], others learn to classify from the instances of the concepts [4], while most of the recent ones combine these techniques [3,5,6].
- ▷ *The type of membership degree they use.* Some use hard thresholds: the subsets $\Phi_1, \ldots, \Phi_n$ are crisp sets, and a pair either belongs to the set or does not [6]. Others implicitly consider these subsets as fuzzy sets, and pair can belong to these sets with different degrees of membership [3].

A more detailed review of these approaches can be found in [10,7].

3 Mapping issues

A matcher analyses only some aspects of the hypothetical relation between two terms, and may lack important information. For example, comparing strings omits the fact that terms have a conventional meaning attached to them. Therefore, it becomes important to combine the results from different matchers, in order to exploit all the available information. To combine the results it is necessary to interpret them in a semantically uniform way. Matchers return different types of results: they can return natural numbers, boolean values, ratios. A possible interpretation, as described in [3], is to consider the result a measure of the plausibility of the correspondence between the terms in a pair.

We have seen in section 2 that to map a term t, a matcher is called to evaluate pairs from $t \times O_{target}$. However, it may often be the case that a matcher cannot distinguish between pairs: for example, EditDistance will return the same result "1" for $\langle rate, race\rangle$,$\langle rate, rave\rangle$, etc. According to the previous subsection, the interpretation is that the pairs must have the same plausibility.

Moreover, results that are near can be interpreted as sharing the same plausibility. For example, it is not meaningful to assign a different confidence to pairs with distance of 5 and 6: both are unlikely to be the mapping. Thus, it is possible to define intervals whose internal values correspond to the same plausibility.

A matcher may also be unable to give evaluation for a pair, as it lacks information: in this case, all hypotheses are equally probable. Matchers may also have different degrees of reliability. The reliability measures how probable is that an assertion made by a matcher is correct [2].

4 A mathematical framework to combine the matchers

There are different mathematical theories that can be used as a framework for a system that must handle the uncertainty issues discussed in section 3, among which the Bayesian approach and Dempster-Shafer are the strongest candidates. Dempster-Shafer [12] has been considered for different uses (medical diagnosis [1], query answering [8]) and it is particularly adapt to tackle them. Using this theory to model the mapping process it is possible to give a uniform interpretation, consistent with the uncertainty inherent in the problem, to the results of the matchers and to combine them in a mathematically sound way.

In Dempster-Shafer the mass is distributed on *sets* of propositions. The mass distribution function $m(\cdot)$ distributes a mass in the interval [0,1] to each element of the power set 2^{Θ} of the set of propositions $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ called *frame of discernment*. The total mass distributed is 1 and the *closed world assumption* is generally made (mass 0 is assigned to the empty set $\emptyset$). The mass $m(\Theta)$ assigned to the frame is the mass that cannot be to assign to any particular subset of Θ. Different mass distributions are combined using *Dempster's rule of combination.*

The model is applied to the function in expression 2 that searches an unknown entity t_j from an ontology O_{target} that best matches a given term t in an ontology O_{source}. The frame of discernment Θ of the problem becomes the Cartesian product $t \times O_{target}$, where each proposition is a pair $\langle t, t_i \rangle$.

Interpreting the results

In Dempster-Shafer, mass assigned to a proposition means support to the belief that the proposition is true. In this model, the matcher is considered an "expert" that gives an opinion about the similarity of terms. The similarity measure must be converted into a measure of the belief in the correctness of the mapping.

As we have seen in section 3, a matcher cannot distinguish pairs of terms that yield the same results and it may be indifferent to pairs with similar results. Therefore, the range of possible results of a matcher is split into intervals. An interval i_k corresponds to a mass m_k: pairs whose results fall into the interval are grouped in the same set s_k, and the belief in the fact that the correct mapping belongs to the set s_k is given by m_k. For example, for EDITDISTANCE the intervals and their masses are $\{\langle[0,0], 0.48\rangle, \langle[1,2], 0.3\rangle, .., \langle[5, ..], 0.0\rangle\}$.

A matcher may lack the information needed to evaluate correctly a pair. In this case, the mass is not allocated, and it should be transferred to the frame of discernment Θ. Matchers can have different degrees of reliability: the mass distributed by a matcher should be discounted by a specific reliability factor. The discounted mass becomes unallocated mass, and should be interpreted as ignorance and transferred to the frame of discernment Θ.

The framework is independent of the matchers used: they are considered plug-in functions that compare pairs. Their results are interpreted using an interface layer, that converts them into mass distributors. A matcher interface *MI* is the tuple $\langle I, \rho \rangle$, where I is the set of intervals $\{\langle r_o, m_1 \rangle, \ldots, \langle r_n, m_n \rangle\}$ of the results range with the corresponding mass, and ρ is the reliability of the matcher used to discount the distributed masses.

The intervals, their masses and the reliability can be computed running the matchers over ontologies with known and validated mappings.

The mass distributions are then combined using *Dempster's rule*:

$$m\left(C\right) = \frac{\sum_{A \cap B = C} m_1(A) m_2(B)}{1 - \sum_{A \cap B \neq \emptyset} m_1(A) m_2(B)} \tag{3}$$

A problem of Dempster's rule is that normalisation can yield counterintuitive results when combining contradictory evidences [13], a common situation when aggregating results from different matchers. A possible solution is to avoid the normalisation. This means to drop the closed world assumption [11] by making $Bel\left(\emptyset\right) \neq 0$ possible.

Choosing the mapping

Once the masses have been distributed and combined, it is necessary to extract the most likely entity from the mass distribution. Dempster-Shafer makes it possible to compute the ***belief*** and the ***plausibility*** about a set $A \subseteq \Theta$:

$$Bel(A) = \sum_{B \cap A} m(B) \qquad Pl(A) = 1 - Bel(\overline{A}) = \sum_{B \cap A \neq \emptyset} m(B)$$

The plausibility forms the upper bound for the belief in A. In some interpretation, the interval $[Bel(A), Pl(A)]$ is the ignorance about A.

In the current framework, belief and plausibility are computed for singletons. The best mapping is chosen ordering the pairs by plausibility, and discarding all the pairs with plausibility and belief below an arbitrary threshold, and with ignorance higher than an arbitrary threshold. This thresholding guarantees that pairs with high plausibility, but low belief are discarded.

5 Testing

The framework described in this paper is independent of the matchers used. However, to test the general concept, the algorithm has been implemented and it is freely available[2]. The tests were executed, with different sets of matchers, comparing two pairs of ontologies, after manually creating the mappings between their entities for comparison. The first pair are ontologies 101 and 205 from the Ontology Alignment Evaluation Initiative[3]. The second pair of ontologies were created for experiments of interaction between agents. The ontologies are available at the project url.

6 Conclusion

In this paper I have discussed the issues that ontology mapping systems must address, and I have proposed a generic framework that allows to combine different matching algorithms. The framework is independent of the actual matchers

---

[2] `http://pyontomap.sourceforge.net`

[3] `http://oaei.ontologymatching.org/2006/`

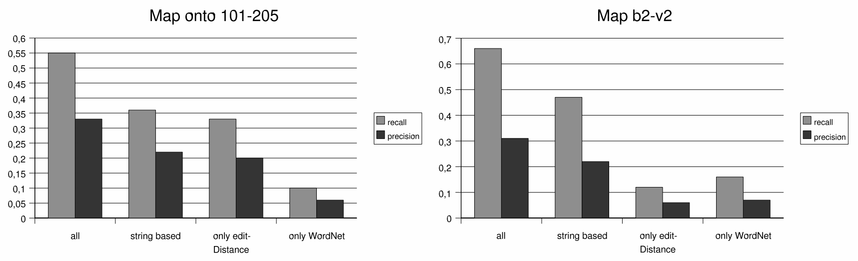


Figure1. Mapping statistics

used. The main result of the framework is to give a consistent interpretation to results returned by different matchers and to provide a mechanism for combining them. The framework's implementation is under development, and uses an *ad hoc* set of matchers, and while the results are still provisional and need improvement, the framework behaviour is consistent with the goals.

References

1. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, chapter 13, pages 272–292. Addison-Wesley, 1984.
2. P Besana, D Robertson, and M Rovatsos. Exploiting interaction contexts in p2p ontology mapping. In *P2PKM*, 2005.
3. Hong Hai Do and Erhard Rahm. Coma - a system for flexible combination of schema matching approaches. In *VLDB*, pages 610–621, 2002.
4. AnHai Doan, J Madhavan, R Dhamankarse, P Domingos, and A Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12(4):303–319, 2003.
5. M Ehrig and S Staab. Qom - quick ontology mapping. In *International Semantic Web Conference*, pages 683–697, 2004.
6. F Giunchiglia, M Yatskevich, and E Giunchiglia. Efficient semantic matching. In *ESWC*, pages 272–289, 2005.
7. Y Kalfoglou and M Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.
8. E Motta M Nagy, M Vargas-Vera. Ontology mapping with domain specific agents in aqua. In *1st Workshop on End User Aspects of the Semantic Web, Heraklion, Crete*, pages 69–83, 2005.
9. S Melnik; H Garcia-Molina; E Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering*, page 117, 2002.
10. P Shvaiko and J Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, 4:146–171, 2005.
11. P. Smets. The combination of evidence in the transferable belief model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):447–458, 1990.
12. R Yager. *Advances in the Dempster-Shafer Theory of Evidence.* John Wiley, New York, 1994.
13. L.A. Zadeh. Review of shafer's a mathematical theory of evidence. *AI-Magazine*, (5):81–83, 1984.

Aligning Multiple Anatomical Ontologies through a Reference

Songmao Zhang[1], Ph.D., Olivier Bodenreider[2], M.D., Ph.D.

[1]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, P. R. China
[2]U.S. National Library of Medicine, NIH, Bethesda, Maryland, USA
[1]smzhang@math.ac.cn, [2]olivier@nlm.nih.gov

Objective: To investigate the feasibility of deriving an indirect alignment between two ontologies from the two direct alignments of these ontologies to a reference ontology. The three anatomical ontologies under investigation are the Adult Mouse Anatomical Dictionary (MA), the NCI Thesaurus (NCI) and the Foundational Model of Anatomy (FMA). **Methods:** The direct alignment employs a combination of lexical and structural similarity. The indirect alignment simply derives mappings from direct alignments to the reference ontology. Each of the three ontologies is used, in turn, as the reference and evaluated against the other two ontologies. **Results:** Number of direct mappings identified: MA-NCI: 715, MA-FMA: 1,353 and NCI-FMA: 2,173. Number of indirect mappings identified through the reference: FMA: 703, NCI: 771 and MA: 741. Mappings specific to direct and indirect alignments are presented and discussed. **Conclusions:** This study confirms the feasibility of aligning two ontologies through a reference ontology. We also show that both the number of concepts and the number of concept names in the reference ontology are important parameters determining the suitability of an ontology to serve as a reference for deriving indirect mappings.

1 Introduction

Mappings among ontologies constitute an enabling resource for applications such as knowledge sharing and application system communication. In particular, such mappings represent a crucial component of the Semantic Web in which the semantic annotation of resources will inevitably draw on multiple ontologies [1]. In previous work, we developed methods for aligning ontologies of anatomy, including the Foundational Model of Anatomy and the Adult Mouse Anatomical Dictionary, as well as the representation of anatomical entities in broader ontologies covering anatomy (GALEN, NCI Thesaurus and SNOMED CT) [2, 3].

While most ontology alignment techniques result in direct, pairwise mappings between ontologies, we have also demonstrated the feasibility of using one ontology as the reference in order to derive indirect mappings between two ontologies themselves mapped to this reference. More specifically, we developed an indirect mapping between the NCI Thesaurus and the Adult Mouse Anatomical Dictionary using the Foundational Model of Anatomy as the reference ontology [4]. The indirect mapping through the FMA was evaluated not only against the direct mapping between NCI and MA, but also against a gold standard alignment established manually between these two ontologies. The main finding of this previous study is that 91% of the matches identified by the direct alignment were present in the indirect alignment. Additionally, a small number of matches not present in the direct alignment were identified indirectly. Such additional matches come from additional synonyms and relations provided by the FMA that are not present in MA or NCI. In contrast, some matches are specific to the direct alignment, i.e., could not be discovered through the indirect alignment. Differing coverage and differing representation were identified as the causes for failure to find these mappings indirectly.

While encouraging, these results also raised the following question. Would we achieve a similar performance if NCI or MA – not FMA – were used as the reference ontology for deriving indirect mappings between the other two ontologies? The objective of this study is to examine this issue, and more generally, to assess the suitability of ontologies to serve as reference in an indirect alignment setting. To this end, we create three variants of our original experiment, using each of the three ontologies, in turn, as the reference to derive indirect matches between the other two.

Ontology matching is an active field of research. It is beyond the scope of this paper to give a detailed account of the various approaches proposed for aligning ontologies. For a survey of such techniques, the interested reader is referred to [5-9]. The most common approach to aligning ontologies is to create direct point-to-point mappings between concepts across ontologies, using a combination of lexical and structural methods (e.g., [10]). However, the role of reference ontologies in ontology alignment is also discussed in the literature. [11] suggests that a better solution for creating, integrating and maintaining multiple local ontologies is to adopt a global reference ontology and a group of mapping rules between them. IF-MAP [12] is an ontology mapping system whose goal is to generate an isomorphism between local ontologies (populated with instances by different communities) and a reference ontologies (unpopulated). In contrast to this approach, we propose to map the "local ontologies" not only to the reference, but also to themselves, *through* the reference. More formally, we use the direct mappings of two ontologies O_1 and O_2 to a reference domain ontology O_R to derive an indirect mapping between O_1 and O_2. [13] proposes a similar approach, but for database integration purposes. Their system builds matchings between local database schemas and a reference ontology, and then composes these matchings to form mappings between schemas. Analogously, TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources) uses ontologies to form a global schema over multiple heterogeneous resources [14]. Here the ontology forms a mechanism for building queries using a common ontological form which is mapped to each of the underlying resources. More recently, both [15] and [16] addressed the related issue of missing background knowledge in ontology matching. The former proposes a fully automatic solution by using semantic matching iteratively, while the latter first aligns the two ontologies with the background ontology, and then uses the structure of background knowledge to derive semantic relationships between the two ontologies.

3 Materials

The **Adult Mouse Anatomical Dictionary** (MA) is a structured controlled vocabulary describing the anatomical structure of the adult mouse [17]. It comprises 2,404 concepts. Each concept has one name (e.g., *Head/neck* and *Adrenal artery*). Additionally, 240 concepts have a total of 259 synonyms (e.g., *Limb* has synonym *Extremity*). The ontology is represented as a directed acyclic graph whose edges represent the relationships *IS-A* and *PART-OF*. Every concept is connected to others through *IS-A* or *PART-OF* relationships. The version used in this study was downloaded on December 22, 2004 (under the name Mus adult gross anatomy in the Open Biomedical Ontologies[1]).

The **NCI Thesaurus** (NCI) provides standard vocabularies for cancer research [18] and its anatomy class describes naturally occurring human biological structures, fluids and substances. The ontology is available in the Ontology Web Language (OWL). There are 4,410 anatomical concepts (accounting for about 12% of all NCI concepts). Every concept has a preferred name (e.g., *Abdominal esophagus*). 1,207 concepts have a total of 2,371 synonyms (e.g., *Orbit* has synonym *Eye socket*). Except for the root (*Anatomic Structure, System, or Substance*), every anatomical concept has at least one *IS-A* relationship to another concept. In addition, anatomical concepts are also connected by a *PART-OF* relationship (named *ANATOMIC STRUCTURE IS PHYSICAL PART OF*). The version used in this study is version 04.09a (September 10, 2004).

The **Foundational Model of Anatomy** (FMA) is an evolving ontology with an objective to conceptualize the physical objects and spaces that constitute the human body [19]. The underlying data model for FMA is a frame-based structure implemented with Protégé. 71,202 concepts cover the entire range of macroscopic, microscopic and subcellular canonical anatomy. In addition to preferred terms, 52,713 synonyms are provided (e.g., concept *Uterine tube* has synonym *Oviduct*). Every concept (except for the root) stands in a unique *IS-A* relation to other concepts. Additionally, concepts are connected by seven kinds of *PART-OF* relationships (e.g., *constitutional part of*, *regional part of*) and their inverses. For the purpose of this study, we considered as only one *PART-OF* relationship (with *HAS-PART* as its inverse) the various kinds of partitive relationships present in FMA. The version used in this study was downloaded on December 2, 2004.

4 Methods

We compare the direct alignment between two ontologies O_1 and O_2 to the indirect alignment automatically generated from mapping both O_1 and O_2 to O_R, the reference ontology. In practice, we perform: 1) three direct alignments O_1-O_2, O_1-O_R and O_2-O_R; 2) the indirect alignment between O_1 and O_2 through their direct alignments with O_R; and 3) a comparison of the direct alignment O_1-O_2 to the indirect alignment obtained through O_R. In [4], the FMA was selected as O_R, and MA

---

[1] http://obo.sourceforge.net/

and NCI as O_1 and O_2, respectively. In the present study, we examine the following two variants: NCI (O_R) with MA (O_1) and FMA (O_2), and MA (O_R) with NCI (O_1) and FMA (O_2).

4.1 Direct Alignment

Our approach to aligning two ontologies directly first compares terms across ontologies lexically in order to identify one-to-one concept matches. The second step is the identification of structural matches. The interested reader is referred to [3] for additional precisions about our method.

The **lexical alignment** compares two ontologies at the term level, by exact match and after normalization. Both preferred terms and synonyms in the two ontologies are used in the alignment. For example, the concepts *Heart valve* in MA and *Cardiac valve* (synonym: *Heart valve*) in FMA are identified as a match. Moreover, synonymy is used to identify additional matches. For example, *Cardiac valve* in NCI and *Heart valve* in MA, although lexically different, are considered a match because they name the same anatomical concept in the Unified Medical Language System® (UMLS®) [20].

The s**tructural alignment** first acquires the inter-concept hierarchical relationships, *IS-A* and *PART-OF*, and their inverses, *INVERSE-ISA* and *HAS-PART*, respectively. Missing relations are generated through complementation, augmentation and inference techniques [3]. Once all relations are represented consistently, the structural alignment is applied to the matches resulting from the lexical alignment in order to identify similar hierarchical paths to other matches across ontologies. For example, the match concepts *Heart valve* in MA and *Cardiac valve* in FMA exhibit similar hierarchical paths to other matches in these two ontologies, including paths to *Heart* (*PART-OF*) and to *Aortic valve* and *Mitral valve* (*INVERSE-ISA*). Such structural similarity is used as **positive evidence** for the alignment. Instead of similar paths, one match may exhibit paths to other matches in opposite directions in the two ontologies. Such paths suggest a structural conflict across ontologies. For example, in MA *Pericardial cavity* stands in a *HAS-PART* relation to *Pericardium*, while in the FMA *Pericardial cavity* is defined as a part of *Pericardial sac*, which is part of *Pericardium*. These conflicts are used as **negative evidence** for the alignment, indicating the semantic incompatibility between concepts across ontologies in spite of their lexical resemblance.

4.2 Indirect Alignment through a Reference

When a concept c_R from O_R is aligned with both a concept c_1 from O_1 ($\{O_1: c_1, O_R: c_R\}$) and a concept c_2 from O_2 ($\{O_2: c_2, O_R: c_R\}$), the concepts c_1 and c_2 are automatically aligned ($\{O_1: c_1, O_2: c_2\}$). The direct alignment MA-FMA identifies the match {MA: *Heart valve*, FMA: *Cardiac valve* (synonym: *Heart valve*)}, supported by positive evidence. The direct alignment NCI-FMA identifies {NCI: *Cardiac valve*, FMA: *Cardiac valve*}, also supported by positive evidence. Therefore, {MA: *Heart valve*, NCI: *Cardiac valve*} is derived automatically, through the FMA concept *Cardiac valve*, and supported by positive evidence in both direct alignments.

5 Results

Results for **three direct alignments** are summarized in section A of **Table 1**. The alignment NCI-FMA yielded the largest number of matches (2,173) and MA-NCI the smallest (715). A very small number of conflicts was identified in the two direct alignments to FMA; none in the direct MA-NCI alignment. In the three direct alignments, a vast majority of the matches (> 90%) was supported by positive structural evidence. No evidence (positive or negative) was found for 5-9% of the matches in three direct alignments. For example, although *Elbow joint* has relations to other matches in both MA (e.g., *PART-OF Forelimb*) and NCI (e.g., *PART-OF Skeletal system*), none of these paths are shared.

Results for the **three indirect alignments** are summarized in section B of **Table 1**. 703 matches between MA and NCI, 771 between MA and FMA, and 741 between NCI and FMA were automatically obtained from using FMA, NCI and MA as a reference, respectively. The majority of the three indirect alignments (88-92%) received positive evidence in both corresponding direct alignments they were derived from. 7-12% of them received no evidence and 0.4-1% received negative evidence in at least one of the direct alignments.

Taking the three ontologies pairwise, we compared the matches obtained in their direct alignment to the matches resulting from their indirect alignment through the reference. The results of these **three comparisons** are summarized in section C of **Table 1**. For **MA-NCI**, 654 matches are shared by both alignments, leaving 61 matches specific to the direct alignment (accounting for 8.5% of the direct matches) and 49 specific to the indirect alignment through the FMA. For **MA-FMA**, 708 matches are shared by both alignments, leaving 645 matches specific to the direct alignment (accounting for 47.7 % of the direct matches) and 63 specific to the indirect alignment through the NCI. For **NCI-FMA**, 710 matches are shared by both alignments, leaving 1,463 matches specific to the direct alignment (accounting for 67.3% of the direct matches) and 31 specific to the indirect alignment through the MA.

88-89% of the shared matches in the three groups received positive structural evidence in all three direct alignments, e.g., {MA: *Heart valve*, FMA: *Cardiac valve*} in MA-FMA. Moreover, about 10-11% of the shared matches in the three groups received no evidence in at least one of the three direct alignments. For example, although linked to other matches in MA (e.g., *HAS-PART Lung*) and FMA (e.g., *HAS-PART Ear*), *Body* has no hierarchical paths to any other matches in NCI. This is why the matches of *Body* received no evidence in the two direct alignments MA-NCI and NCI-FMA, while receiving positive evidence in direct alignment MA-FMA. Lastly, nearly 1% of the shared matches in the three groups received negative evidence in one of the three direct alignments. For example, although a concept *Nephron* exists in the three ontologies, the corresponding match received negative evidence in the direct MA-FMA alignment (i.e., links to *Renal tubule* (synonym: *Uriniferous tubule*) through *HAS-PART* in MA but links to *Uriniferous tubule* through *PART-OF* in FMA), while receiving positive evidence in both direct alignments MA-NCI and NCI-FMA. Domain knowledge is required to evaluate such matches.

Table 1. Number of matches in the direct and indirect alignments

		MA – NCI	**MA - FMA**	**NCI - FMA**
A	Direct alignment	715 matches (91.3% positive evi.)	1,353 matches (94.8% positive evi.)	2,173 matches (90.1% positive evi.)
B	Indirect alignment	FMA as reference	NCI as reference	MA as reference
		703 matches (92% positive evi.)	771 matches (88.1% positive evi.)	741 matches (87.6% positive evi.)
C	Shared by direct & indirect alignment	654 matches	708 matches	710 matches
	Specific to direct alignment	61 matches	645 matches	1,463 matches
	Specific to indirect alignment	49 matches	63 matches	31 matches
D	Shared / direct alignment	91.5%	52.3%%	32.7%

6 Discussion

Alignment through a reference ontology is feasible and efficient. This study confirms the feasibility and efficiency of aligning two ontologies through a reference ontology. The proportion of matches from the direct alignment also identified in the indirect alignment is particularly good (91.5%) in the alignment MA-NCI with FMA as the reference. Assuming a good reference ontology is available, alignment through a reference is cost-effective: aligning *n* ontologies requires *n(n-1)/2* pairwise mappings, but only *n-1* mappings to a reference ontology. For five ontologies – which is a small number by Semantic Web standards – the difference already represents a 60% economy (4 vs. 10).

Suitability as a reference Ontology: Size matters. As shown in section D of **Table 1**, using the FMA as a reference resulted in the identification of a vast majority (91.5%) of the direct matches between MA and NCI. The large size of the FMA and its comprehensive set of synonyms contributed to this high percentage of mappings [4]. In contrast, when using NCI or MA as the reference in indirect alignment, only a fraction of the direct matches could be identified. Only one half (52.3%) of the corresponding direct matches were identified through the NCI and one-third (32.7%) through the MA as a reference. These findings confirm our intuition that ontologies offering a small number of concepts and a limited number of names for each concept are less suitable as a reference for deriving an indirect alignment between two ontologies. In the case of MA, for example, there are only 2,404 concepts and 2,663 names in comparison to over 70,000 concepts and 120,000 names in the FMA.

Every ontology, large or small, contributes specific indirect matches. Regardless of its size, as shown in section C of **Table 1**, every ontology contributes specific indirect matches, i.e., matches that are not identified in the direct alignment. For example, using MA as a reference generated 31 specific matches, of which 19 received positive evidence in both direct alignments. For example, *Glomerular capillary* in NCI was not mapped directly to *Glomerulus* in FMA because the two terms are not synonyms in either ontology. However, the match {NCI: *Glomerular capillary*, FMA: *Glomerulus*} was identified indirectly when using the MA as a reference because *Glomerulus* and *Glomerular capillaries* are synonyms in MA. The match also received positive evidence in both direct alignments MA-NCI and MA-FMA. This indicates that the MA synonyms, although in relatively small number, play a significant role in the identification of mappings across two larger ontologies.

In summary, the most important finding of this study is that deriving an indirect alignment through a reference ontology is not only feasible, but also reasonably efficient. Moreover, this study confirms the intuition that both the number of concepts and the number of concept names in the reference ontology are important parameters determining the suitability of an ontology to serve as a reference for deriving indirect mappings. These findings are compatible with Burgun's "desiderata for domain reference ontologies in biomedicine", including good lexical coverage, good coverage in terms of relations and compatibility with standards [21].

Acknowledgements. This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM) and by the Natural Science Foundation of China (No.60496324), the National Key Research and Development Program of China (Grant No. 2002CB312004), the Knowledge Innovation Program of the Chinese Academy of Sciences, and MADIS of the Chinese Academy of Sciences, and Key Laboratory of Multimedia and Intelligent Software at Beijing University of Technology. Thanks for their support to Cornelius Rosse, José Mejino and Todd Detwiler for the Foundational Model of Anatomy. Terry Hayamizu from the Jackson Laboratory contributed the evaluation of the direct mapping between NCI and MA.

References

1. Uschold, M., Gruninger, M.: Creating semantically integrated communities on the world wide web. Proc. International Workshop on the Semantic Web (2002)
2. Bodenreider, O., Hayamizu, T.F., Ringwald, M., de Coronado, S., Zhang, S.: Of mice and men: Aligning mouse and human anatomies. Proc AMIA Symp (2005) 61-65
3. Zhang, S., Bodenreider, O.: Aligning representations of anatomy using lexical and structural methods. AMIA Annu Symp Proc (2003) 753-757
4. Zhang, S., Bodenreider, O.: Alignment of multiple ontologies of anatomy: Deriving indirect mappings from direct mappings to a reference. Proc AMIA Symp (2005) 864-868
5. Doan, A., Halevy, A.Y.: Semantic integration research in the database community: A brief survey. AI Magazine **26** (2005) 83-94
6. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. Knowledge Engineering Review **18** (2003) 1-31
7. Noy, N.F.: Semantic integration: a survey of ontology-based approaches. SIGMOD Rec. **33** (2004) 65-70
8. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal **10** (2001) 334-350
9. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics **4** (2005) 146-171
10. Noy, N.F.: Tools for mapping and merging ontologies. In: Staab, S., Studer, R. (eds.): Handbook on Ontologies. Springer-Verlag (2004) 365-384
11. Uschold, M.: Creating, integrating and maintaining local and global ontologies. Proceedings of the Workshop on "Applications of ontologies and problem-solving methods" at the 14th European Conference on Artificial Intelligence (ECAI 2000) (2000) 20-25
12. Kalfoglou, Y., Schorlemmer, M.: IF-Map: an ontology mapping method based on information flow theory. In: Spaccapietra, S. (ed.): Journal on Data Semantics (LNCS 2800), Vol. 2800 (2003) 98-127
13. Dragut, E., Lawrence, R.: Composing Mappings Between Schemas Using a Reference Ontology. In: Meersman, R., Tari, Z. (eds.): On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2004, Agia Napa, Cyprus, October 25-29, 2004. Proceedings, Part I (LNCS 3290), Vol. 3290. Springer, Berlin / Heidelberg (2004) 783-800
14. Goble, C.A., Stevens, R., Ng, G., Bechhofer, S., Paton, N.W., Baker, P.G., Peim, M., Brass, A.: Transparent Access to Multiple Bioinformatics Information Sources. IBM Systems Journal Special issue on deep computing for the life sciences **40** (2001) 532-552
15. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: Discovering missing background knowledge in ontology matching. In: Brewka, G., Coradeschi, S., Perini, A., Traverso, P. (eds.): Frontiers in Artificial Intelligence and Applications - Proceedings of the 17th European Conference on Artificial Intelligence (ECAI), Vol. 141. IOS Press (2006) 382-386
16. Aleksovski, Z., Klein, M., ten Kate, W., van Harmelen, F.: Matching unstructured vocabularies using a background ontology. In: Svatek, S.S.a.V. (ed.): Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW'06) (2006) (in press)
17. Hayamizu, T., Mangan, M., Corradi, J., Kadin, J., Ringwald, M.: The Adult Mouse Anatomical Dictionary: a tool for annotating and integrating data. Genome Biology **6** (2005) R29
18. De Coronado, S., Haber, M.W., Sioutos, N., Tuttle, M.S., Wright, L.W.: NCI Thesaurus: Using Science-based Terminology to Integrate Cancer Research Results. Medinfo **2004** (2004) 33-37
19. Rosse, C., Mejino, J.L., Jr.: A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J Biomed Inform **36** (2003) 478-500
20. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res **32 Database issue** (2004) D267-270
21. Burgun, A.: Desiderata for domain reference ontologies in biomedicine. J Biomed Inform **39** (2006) 307-313

Combining Ontology Mapping Methods Using Bayesian Networks

Ondřej Šváb, Vojtěch Svátek

University of Economics, Prague, Dep. Information and Knowledge Engineering,
Winston Churchill Sq. 4, 130 67 Praha 3, Prague, Czech Republic
`svabo@vse.cz, svatek@vse.cz`

Abstract. Bayesian networks (BNs) can capture interdependencies among ontology mapping methods and thus possibly improve the way they are combined. Experiments on ontologies from the OAEI collection are shown, and the possibility of modelling explicit mapping patterns in combination with methods is discussed.

1 Introduction

Most existing systems for ontology mapping combine various methods for achieving higher performance in terms of recall and precision. Our approach relies on *Bayesian networks* (BNs) as well-known formal technique that can capture interdependencies among random variables. A Bayesian network (BN) [3] is a directed acyclic graph with attached local probability distributions. Nodes in the graph represent random variables with mutually exclusive and exhaustive sets of values (states). Edges in the graph represents direct interdependences between two random variables. We believe that this approach can bring additional benefits compared to ad hoc combination of methods, mainly resulting from better adaptability (training from data within a well-established formal framework).

Two approaches that use BNs for Ontology Mapping have recently been reported. The first is *OMEN* [4], which mainly serves for enhancing existing mappings. Its input are results of another mapping tool, while its output are more precise mappings as well as and new mappings. Nodes in the BN represent pairs of concepts that can potentially be mapped. Edges follow the taxonomy given in original ontologies. The network structure thus mimics that of ontologies themselves, though heuristics for graph pruning are employed in this transformation. For constructing conditional probability tables (CPTs) for each node meta-rules are used, such as : "if two nodes match and so do two arrows coming out of these nodes then the probability that nodes at the other end of the arrows match is increased". The second project, *BayesOWL* ([5]), is rather a framework for ontology mapping than a mapping method per se. The probabilistic ontological information is assumed to be learnt (in forms of probabilistic constraints) from web data using a text-classification-based learner; this information is translated to BNs. Mappings among concepts from two different ontologies then can be discovered using so-called evidential reasoning across two BNs.

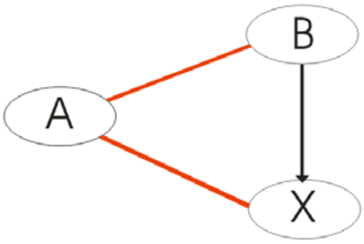


Fig. 1. Example of mapping pattern across two ontologies

2 Modelling Dependencies among Mapping Methods

Our approach differs from prior approaches in the sense that we don't apply BN modelling to ontologies or their mappings themselves but rather to different *mapping methods*. The BNs are assumed to contain nodes (or sub-networks) representing the results of individual methods plus one representing the final output. This will allow us not only to combine the methods (in the probabilistic framework) but also to talk about conditionally in/dependent methods, a minimal required subset of methods and the like. The mapping methods can have varying degree of granularity: we focus on low-level methods, understood as *mapping justifications*. Moreover, in the work-in-progress part of our research, we account for *mapping patterns* encompassing small structural fragments of ontologies. The patterns will capture, to some degree, similar information as OMEN meta-rules, we however prefer to model them directly within the BN formalism.

We distinguish among *families of methods* (string-based, linguistic-resource-based, graph-based, logic-based etc.) sharing some generic principle and input resources. Each family encompasses multiple low-level methods; for example, a string-based method can be built upon diverse string distance measures. We dedicate a separate node of the BN to each low-level method, viewed here as *mapping justification*. We believe that such methods are a meaningful target for BN modelling, as their statistical dependencies are likely to reflect plausible relationships even interpretable by a human.

The notion of *mapping pattern* is a natural counterpart to that of intra-ontology ('design') pattern [1]. Mapping patterns have been implicitly proposed by Ghidini & Serafini [2], who even consider mappings among different modelling constructs (such as concept-to-relation). A mapping pattern is, essentially, a structure containing some (at least one) constructs from each of the two (or more) ontologies plus some (candidate) mapping among them. The simplest mapping pattern only connects one concept from each of the two ontologies. An example of a bit more complex mapping pattern is in Figure 1. The left-hand side (class A) is from O_1 and the right-hand side (class B and its subclass X) is from O_2. We try to map class A simultaneously to class B and to class X.

The input to the process of BN *training* for ontology mapping are positive and negative examples with results of individual methods ('mapping justifications'), and possibly also the network structure, unless we want to learn it as well. The positive examples correspond to pairs for which mapping has previously been established, while the negative ones are (all or a subset of) pairs that have been identified as non-matching. Then CPTs and possibly the structure are learnt. In the phase of using the trained BN, the mapping justifications for unseen cases (pairs of concepts) are counted and inserted into the BN as evidence. The result of alignment is calculated via propagation of this evidence.

3 Experiments

For experiments we choose ontologies from the *OntoFarm* collection (`http://nb.vse.cz/~svabo/oaei2006/`), which is currently part of the OAEI 2006 setting. It models the domain of conference organisation; individual ontologies were designed independently by different people and based on different resources: personal experience with conference organisation, conference web pages or conference organisation support tools.

We restricted the first experiments to ten string distance measures implemented in the *SecondString* library (`http://secondstring.sourceforge.net/`: Levenshtein, Jaro, Jaccard, Char-Jaccard, Smith-Waterman, Monge-Elkan, SLIM, TokenFelligiSunter, UnSmoothedJS and TFIDF. Because of the local nature of distance string measures, capturing context by means of mapping patterns does not seem to bring great benefits; we thus only focused on the combination of low-level methods. We extracted classes from two ontologies (`ekaw.owl` and `ConfOf.owl`). Our training data consist of 798 pairs, of which 149 were manually labelled as positives and 649 as negatives. They were 'semi-randomly' picked from different parts of the ontology; the overall number of possible pairs would be about 2500 (the product of concept counts in both ontologies). The results were transformed from the $[0, 1]$ scale to two categories: 'true' if the value is over 0.5 and 'false' if the value is lower or equal to 0.5.

To learn the BN we use the *Hugin* tool (`http://www.hugin.com/`): the structure was trained using the NPC method and CPTs were trained using the EM algorithm. We learnt two Bayesian networks in this way. The first one has been enforced the *naive Bayesian structure*, which assumes independence of methods; only the CPTs were learned from data. For the second network, we also *learnt the structure*; in this way we could also explore interdependencies among low-level methods. The learnt structure is in Figure 2. From the structure and the defintion of so-called *Markov blanket* [3] we can conclude that if we know the mapping justifications of TFIDF, Smith-Waterman, Jaccard, Jaro, and SLIM, other methods do not matter. Methods unrelated to some other method (TokenFelligiSunter and UnSmoothedJS) are not in the BN at all.

To evaluate the performance of each proposed Bayesian classifier we used the *one-leave-out method.* For the naive Bayesian classifier, we got the best result with probability threshold 80%: 73% precision, 60% recall (F-measure was then

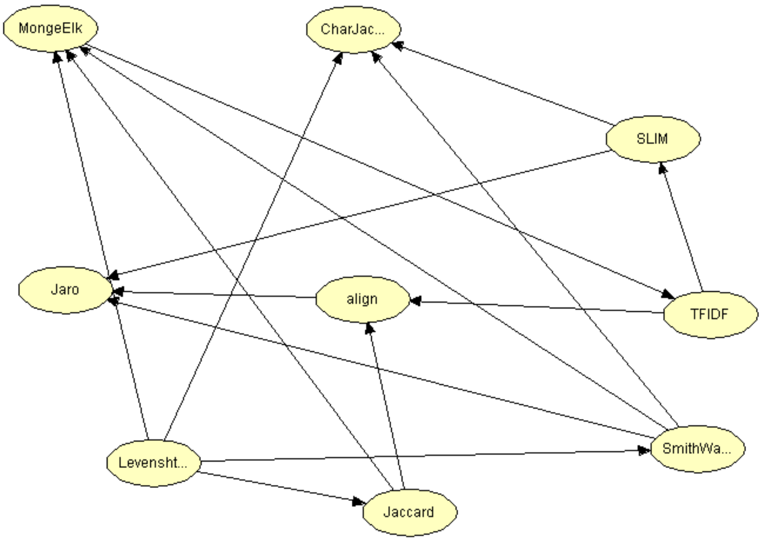


Fig. 2. BN - automatically learnt structure

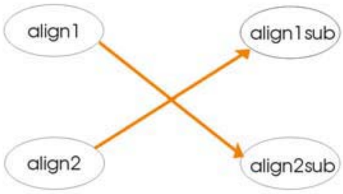


Fig. 3. Fragment of BN reflecting the mapping pattern from Fig. 1

0.66) and 88% accuracy. For the Bayesian classifier with learnt structure we got 84% precision, 53% recall (F-measure was 0.65) and 89% accuracy as best result, for whatever threshold between 40% and 70%. Both our classifiers outperform trivial classifiers that always predict true or false, respectively. Overall, the Bayesian classifier with learnt structure outperformed the naive Bayesian classifier. On the other hand, the best individual method (Jaccard) performed the same as the Bayesian classifier with learnt structure (84% precision, 53% of recall and 89% accuracy) with threshold around 50%. By this result, we can say that the combination (using BN) of string distance measures does not bring a direct benefit. However, the (second) Bayesian classifier is less sensitive to the change of threshold, while Jaccard moves towards 100% precision but rather low recall of 23% as soon as the threshold increases to 60%.

4 Conclusions and Future Work

We suggested to use low-level methods as 'mapping justifications' in order to train a Bayesian network on a sample of mappings to produce new mappings. Results of preliminary experiments with string distance measures as low-level methods are not entirely convincing in terms of performance, which can be explained by strong correlation among these methods; this correlation was actually discovered when learning the BN structure. The main role of this initial phase of research was to gain deeper insight into the problems addressed. The possibility to model explicit mapping patterns in combination with methods was also studied but not yet reflected in experiments.

In the future, we plan to employ, in the role of *mapping justifications*, not only string-based (low-level) techniques, but also e.g. graph-based or thesauri-based techniques. A more challenging task is however to design BNs reflecting the structure of *patterns*. Each method (and the final result) will be represented with a *set of nodes* corresponding to the given pattern. For example, a fragment of BN reflecting the mapping pattern from Fig. 1 is depicted in Fig. 3. It considers not only the equivalence relation but also the (proper) subsumption relation, and has four nodes that represent the alignment of each pair and each relation (equivalence of A and B, equivalence of A and X, subsumption of A and B and subsumption of A and X). `align1` represents the equivalence mapping between A and B. `align1sub` represents the subsumption mapping between A a B ($B \supset A$). `align2` represents the equivalence mapping between A a X. Finally, `align2sub` represents the subsumption mapping between A a X ($A \supset X$). Edges then should automatically be learnt for the pairs of nodes `align1` and `align2sub`, and `align2` and `align1sub`, respectively, due to strict dependencies.

We thank Jiří Vomlel for his assistance with Bayesian Networks. The research was partially supported by the IGA VSE grants no.26/05 "Methods and tools for ontological engineering", no.12/06 "Integration of approaches to ontological engineering: design patterns, mapping and mining", and by the Knowledge Web Network of Excellence (IST FP6-507482).

References

1. W3C Semantic Web Best Practices and Deployment Working Group. Ontology Engineering and Patterns Task Force (OEP). Online at `http://www.w3.org/2001/sw/BestPractices/OEP/`
2. Ghidini C., Serafini L.: Reconciling concepts and relations in heterogeneous ontologies. In: Proc. ESWC 2006, Budva, Montenegro, 2006.
3. Jensen F. V.: Bayesian Networks and Decision Graphs. Springer, 2001.
4. Mitra P., Noy N. F., Jaiswal A. R.: OMEN: A Probabilistic Ontology Mapping Tool. In: Workshop on Meaning coordination and negotiation at the Third International Semantic Web Conference (ISWC-2004), Hiroshima, Japan, 2004.
5. Pan R., Ding Z., Yu Y., Peng Y.: A Bayesian Network Approach to Ontology Mapping. In: Proceedings ISWC 2005.

Discovering and Merging Keyword Senses using Ontology Matching[⋆]

Mauricio Espinoza[⋆⋆], Raquel Trillo, Jorge Gracia, and Eduardo Mena

IIS Department, Univ. of Zaragoza, María de Luna 1, 50018 Zaragoza, Spain
`{mespinoz,raqueltl,jogracia,emena}@unizar.es`
WWW home page: `http://sid.cps.unizar.es`

Abstract. During the last years we are witnessing how the use of keywords has become the standard input when searching the Web. As opposite to the syntactic searches performed by traditional web search engines, the current research challenge is a semantics-guided information retrieval. The increasing pools of ontologies available on the Web can help to discover the semantics of user keywords and this information is priceless for many tasks, including new semantic search engines.
In this paper we propose a system that takes as input a list of keywords provided by the user and discovers their possible meanings by consulting the knowledge represented by many (heterogeneous and distributed) ontologies. These keyword senses are semantically enriched with the synonym terms found during the ontology matching process: A synonymy measure based on statistics techniques and ontological similarity is used to integrate senses that are similar enough.

Keywords: Ontology matching for information integration

1 Introduction

Although keyword-based search is a widely used technique for information retrieval, traditional techniques do not consider the specific semantics assigned by the user: the same keywords can be used by different users with the purpose of accessing to different information. Furthermore, the syntactic-based search engines are very influenced by the enormous amount of information about popular issues on the Web, i.e., the keyword "java": Java as programming language eclipses the rest of possible senses (the Indonesian island, a coffee plant, different US cities, etc). However, ontologies (which offer a formal, explicit specification of a shared conceptualization [5]) can be used to make the semantics of user keywords explicit without ambiguity. The more ontologies consulted, the more chances to find the semantics assigned to keywords by the user.

In this paper, we propose a system that takes as input a list of plain keywords provided by the user, discovers their semantics in run-time and obtains a

---

[⋆] This work is supported by the CICYT project TIN2004-07999-C02-02.

[⋆⋆] Work supported by a grant of Santander Central Hispano & University of Zaragoza.

list of senses extracted from different ontology pools; it deals with the possible overlapping among senses. The main steps of our approach are summarized in the following:

1. *Extraction of Keyword Senses.* First, the user keywords are normalized by a preprocessing step (e.g., rewriting them in lowercase, removing hyphens, etc.), and in order to discover the semantics of the user keywords, the system accesses to the shared knowledge stored in different ontology pools available on the Web. The extracted senses are semantically enriched with the ontological senses of their synonyms (which are obtained from the ontology pool), whenever the system evaluates that the synonym senses matches to the semantics of the corresponding keyword sense.
2. *Alignment of Senses.* This process uses an incremental algorithm for the alignment of the different keyword senses in order to remove the possible semantics redundancy among them. Senses are merged when the estimated synonymy probability between them is above a certain threshold. The synonymy measure combines a standard string distance metric with a structural similarity measure that is based on vector space techniques. Thus the result is a set of *different* possible senses for each user keyword.

For efficiency purposes, the system uses sampling and other statistic techniques, as well as parallel processing, whenever possible. The output of our system can be the input for a disambiguation process across keywords [4] or used to retrieve data once the keyword semantics is known.

The rest of this paper is as follows. In Section 2 we show how the possible senses of each keyword are obtained and semantically enriched with their synonym senses. In Section 3 we describe the algorithm that computes the synonymy probability in order to integrate senses when a certain threshold is achieved. Finally, conclusions and future work appear in Section 4.

2 Extraction of Keyword Senses

In this section we provide the details that show the contribution of this paper in the task of automatically retrieving the possible senses for a set of user keywords. In order to find the ontological terms that match those keywords, the system accesses to Swoogle [2], other remote lexical resources as WordNet [8] and other ontologies not indexed by Swoogle are used as well. We advocate using a pool of ontologies instead of just a single one, like WordNet (as many works do [6, 7]), because many technical or subject-specific senses cannot be found in WordNet.

The system builds a sense for each URI obtained with the information retrieved from matching terms in the ontology pool [1]. In our approach, a sense of a keyword k, denoted by s_k, is a tuple $s_k = <s, grph, descr, pop, syndgr>$, where s is the list of synonym names[1] of keyword k, $grph$ describes the sense

---

[1] To extract from an ontology the synonyms of a class, property or individual, the primitives *equivalentClass*, *equivalentProperty* and *sameIndividualAs* are used, respectively.

s_k by means of the hierarchical graph of hypernyms and hyponyms of synonym terms found in one or more ontologies, *descr* is a description in natural language of such a sense, and *pop* and *syndgr* measure the degree of popularity of this sense (*pop* is the number of times it appears in the ontology pool and *syndgr* is the integrated percentage of synonymy degree). Thus, senses are built with the information retrieved from matching terms in the ontology pool [1].

As matching terms could be ontology classes, properties or individuals, three lists of possible senses are associated with each keyword k: S_k^{class}, S_k^{prop} and S_k^{indv}. In Figure 1 we show an example of some senses found in the ontology pool for the user keyword "star". The system finds in WordNet two matchings of keyword "star" as concept/class ($s1$ and $s2$), and one matching in the Travel Ontology[2] as property of class "hotel" ($s3$). Notice that each sense is initialized with a popularity=1 and a synonymy degree=1.

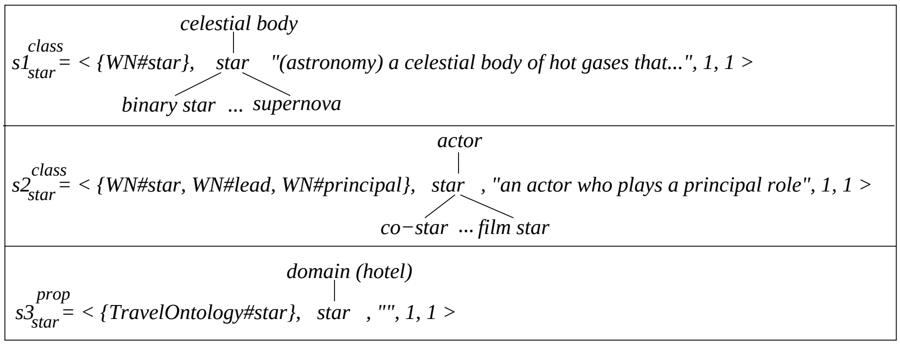


Fig. 1. Some senses of keyword "star" extracted from the ontology pool

Each keyword sense is enhanced incrementally with the synonyms terms extracted from the ontology pool. Therefore our system takes advantage of the shared ontologies available on the Web and semantically enriches the keyword senses with senses extracted from their synonyms. The synonym names are stored in the sense structure shown before, which gets upgraded everytime the sense is integrated with a (very similar) sense coming from other ontology. In order to evaluate the semantic similarity between the sense of a keyword and their synonyms, the system performs a *sense alignment* process (detailed in Section 3) which determines whether the semantics of the keyword sense and each synonym sense found represent the same semantic or not. After discarding the synonym senses that do not enrich the corresponding keyword sense, the result is a list of different possible senses for each keyword.

This process can be limited in time; obtaining the senses is executed in parallel for each keyword; within that task, the semantic enrichment of each keyword sense with its synonym senses is performed in parallel too.

---

[2] http://learn.tsinghua.edu.cn:8080/2003214945/travelontology.owl

3 Alignment of Senses

We explain in this section the sense alignment process, which is used in two situations by our system: 1) to check which synonym senses represent the same semantics as their keyword senses, and 2) to avoid redundancy in the list of possible senses of each user keyword. However both tasks share a common goal: to find when two given senses represent very similar semantics; in that case they will be considered synonyms and both senses will be integrated[3].

In order to decide if two senses must be integrated (as a single sense) or not, the system computes their *synonymy probability.* Thus the system avoid redundancy among the possible senses of a keyword. At present, several solutions to determine the matching among ontological terms of distinct ontologies have been proposed, see [9] for recent surveys. Our approach computes coefficients of synonymy degree in the [0,1] range, however other approaches as semantic matching [3] can be used as well.

The synonymy measure used relies on both *linguistic and structural* characteristics of ontologies. The following steps are performed: 1) an initial computation using linguistic similarity, which consider labels as strings, 2) a recursive computation using structural similarity, which exploits the semantics of terms (ontological context) until a certain depth, and 3) the above values are combined to obtain the resultant synonymy measure.

Our proposal for sense alignment is not just a comparison between two senses but an iterative process, which improves the quality and efficiency of ontology matching and enables the reuse of new discovered senses. In other words, each new integrated sense must be considered as candidate to integrate with the rest. For the same reason, new senses that do not integrate are stored because they could become the missing semantic gap between two senses. Although this method is costly (we limit its execution time), it performs a much better ontology alignment among senses. Due to space limitations, we do not detail this process.

In a variety of approaches, the similarity measure is only calculated among ontological terms that plays as classes. However, unlike another works, we propose a way to obtain the synonymy probability according to the type of senses that we compare. Details about this process is available in [1] as it not the main goal of this paper.

4 Conclusions

In this paper we have presented a semantics-guided approach to discover the possible senses for a set of keywords, by searching and extracting relevant knowledge from different ontology pools; ontology matching and synonymy estimation techniques are used to merge senses considered similar enough. The main features of our proposal are the following:

---

[3] The integration process that we propose can be found in [1].

1. It uses an iterative approach to retrieve from different knowledge repositories the possible senses of each user keyword, in a parallel manner. A sense is represented basically as the (multi)ontological context of a term, and the system is able to deal with senses corresponding to different kind of ontology terms (classes, properties, and individuals).
2. It considers not only the senses corresponding to ontology terms syntactically matching the user keywords but also the senses of ontology terms matching the synonyms of the user keywords, recursively, in order to semantically enrich the keyword senses retrieved within a certain synonymy threshold.
3. It measures the synonymy degree between two senses by considering their linguistic and structural similarity. Statistical techniques like sampling and parallel processing are used to improve the performance of this process.

We believe that this technique to find out the semantic different between senses (subsets of ontologies) can be applied to many fields. As example, we are currently working on using the retrieved senses to generate queries expressed in a knowledge representation language to retrieve data corresponding to the user keywords.

References

1. M. Espinoza, J. Gracia, R. Trillo, and E. Mena. Discovering the semantics of keywords: An ontology-based approach. In *The 2006 International Conference on Semantic Web and Web Services (SWWS'06), Las Vegas, Nevada (USA).* CSREA Press, June 2006.
2. T. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng. Swoogle: Searching for knowledge on the Semantic Web. In *AAAI 05 (intelligent systems demo)*, July 2005.
3. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Discovering missing background knowledge in ontology matching. In *Technical Report DIT-06-005, Informatica e Telecomunicazioni, University of Trento*, February 2006.
4. J. Gracia, R. Trillo, M. Espinoza, and E. Mena. Querying the web: A multiontology disambiguation method. In *Sixth International Conference on Web Engineering (ICWE'06), Palo Alto (California, USA).* ACM, July 2006.
5. T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation.* Kluwer Academic Publishers, 1993.
6. B. Le, R. Dieng-Kuntz, and F. Gandon. On ontology matching problems - for building a corporate semantic web in a multi-communities organization. In *ICEIS (4)*, 2004.
7. V. Lopez, E. Motta, and V. Uren. Poweraqua: Fishing the semantic web. In *3rd European Sematic Web Conference, Budva, Montenegro*, June 2006.
8. G. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), nov 1995.
9. P.Shvaiko and J. Euzenat. A survey of schema-based matching approaches. In *Journal on Data Semantics*, 2005.

A Peer-to-Peer Based Semantic Agreement Approach for Information Systems Interoperability

I Wayan Simri Wicaksana and Kokou Yétongnon

Laboratoire d'Electronique, Informatique et Image (LE2I)
Université de Bourgogne
BP 47870 - 21078 Dijon Cedex, France
`{iwayan, kokou}@u-bourgogne.fr`

Abstract. This paper focuses on P2P based data management and semantic mediation. We propose an approach based on a P2P for semantic interoperability of information sources that aims to combine the advantages of semantic mediation and peer-to-peer systems. It is based on a pure P2P with super peer architecture consisting of two types of peers. The super peer contains a reference ontology, which provides a common ontology (CO) of the domain. The peer contains export schema (ES), which represent local data. The approach based on semantic agreement between CO and ES, which called half agreement (HA). The half-agreements utilize for discovery sources and exchange information among peers.

1 Introduction

Effective information and services sharing in distributed such as P2P based environments raises many challenges, including discovery and localization of resources, exchange over heterogeneous sources, and query processing. Traditional approach does not scale well when applied in dynamic distributed environments and has many drawbacks related to the large numbers of sources.

Several applications of P2P networks can be distinguished, ranging from content sharing applications (e.g. Napster, Gnutella) to distributed computing applications (e.g. SETI@home, Avaki) and development support platforms (e.g. JXTA). Generally, two main categories of P2P systems can be distinguished. Unstructured P2P systems organize peers in networked spaces. Each peer controls and maintains its shared data. User queries are based on (1) centralized directory models where one or more servers are used to record and locate data, or (2) a query routing model that essentially floods the network to determine relevant peers that are likely to contain the requested data. By contrast, structured P2P architectures organize data in a key spaces divided into segments. User queries are based on a Distributed Hashing Table (DHT) built on the top of the overlay structure of peers.

Survey on schema matching [1, 2] explained the general approach of schema matching based on terminology, structural and semantic. Kolfoglou [3] delivered state of the art ontology mapping, which consider some component in framework, methods, tools, translators, mediators, techniques and theoretical.

We propose an approach based on a P2P for information interoperability that aims to combine the advantages of semantic agreement and peer-to-peer systems. Main our contribution is how to create and implement peer semantic agreement for discovery process. The main difference of our approach to Remindin [4] and expertise-based [5] is in calculation of related peers based on similarity of half-agreement (semantic agreement) by calculating *BindingValue*.

The paper is organized as follows: section 2 presents the peer agreement based semantic approach. Section 3 presents an example. Finally, section 4 concludes the paper.

2 Peer Agreement Approach

2.1 Overview

A P2P system $\prod = < \mathcal{P}, \mathcal{A} >$ is a set $\mathcal{P} = \{P_1, ..., P_n\}$ of peers and a set $\mathcal{A}$ of agreements. Two types of peers can be distinguished in the approach. First, Super peers (SP) are used to maintain common ontology of a community. Peers (PP) represent another type of peers that provide or search information.

Figure 1 depicts the general processes of information exchange at P2P as follow: (1) publish: peers can publish their description of the features of the data. In our approach, the publishing will introduce with preprocessing which called half agreement. (2) request: a peer send a request to find appropriate sources for his query. The searching based on relevant advertisement among the currently available peers. The peer can broadcast his half agreement to candidate peers and calculate similarity concept between query and sources. (3) bind: interest parties can create mapping composition based on their half agreement.

2.2 Semantic Similarity of concepts

Our approach utilizes available approaches that based on:

- Label Matching, a label has a part value of semantic, which presented at taxonomy model such as WordNet [6]. There are two steps at label matching [7]. First, a language preprocessing step is used to transform the labels into words prior to linguistic analysis. For example, this step can be used to expand abbreviations and reduce article such as the, a. Next, the labels are matched by determining relations between them. This can be done based on WordNet relations. The WordNet [8] is a brad coverage lexical network of English words. Wu-Palmer (WUP) and Jiang-Conrath (JNC) methods are utilized for the WordNet and combined with threshold value.
- Internal structure, a 'language' attribute [9, 10] is property label of the language such as owl:cardinality, rdfs:label. The similarity value between two entities is derived by the ratio between numbers of similar properties over the maximal number of both entities.

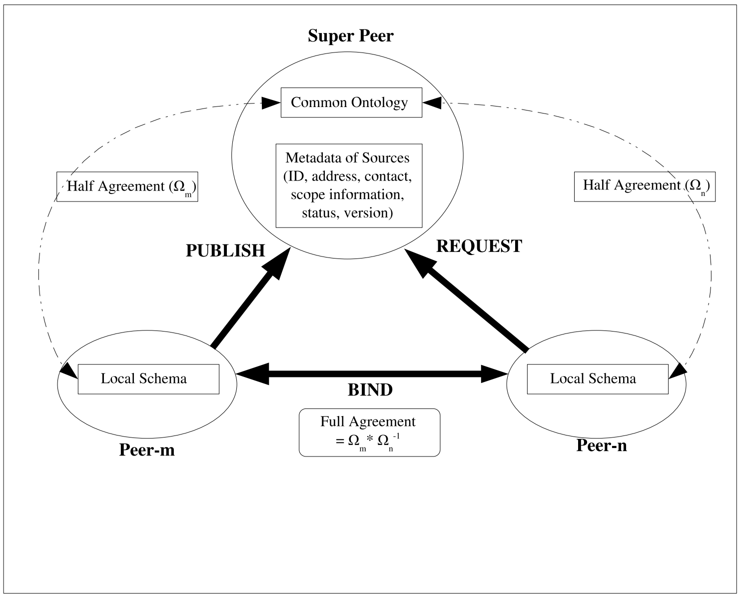


Fig. 1. Publish, Request and Bind in P2P

- The external structure takes into account the position of a concept in a hierarchy. The method refers to upward cotopic distance [11] which compares the similarity of the set of superclasses.

2.3 Semantic Agreement

An agreement unit defines one-to-one or one-to-many mappings between (C^i_{CO}, $\{C^j_{ES}\}$) where C^i_{CO} is an ontology and C^i_{ES} is an export schema concept. An agreement unit encapsulates three main components that are described by RDF/ OWL schemas: (1) an ontology concept, (2) a fragment of an export schema, and (3) the logical mapping function that link the two components. Set of agreement unit is called half-agreement. Full agreement is a composition of half agreement between two peers. A half-agreement unit is represented as tuple:

$$< SMC_{ID}, \{CO^m_{SP}, type^m_{SP}\}, \{ES^n_{PP}, type^n_{PP}\}, \mu^{ID} > \tag{1}$$

where SMC_{ID} is a unique agreement identifier; m=1..m_{max}, is the number of concepts of a Super Peer; CO^m_{SP} is the m-th concept of the super peer; $type^m_{SP}$ is the type of CO^m_{SP} which can be class or property; ES^n_{PP} is the n-th concept of the export schema of PP; $type^i_{PP}$ is the type of ES^n_{PP} which can be a class or a property; n=1...n_{max} is the number of concepts in the peer; μ^{ID} is a logical mapping function for resolving semantic heterogeneities between the super peer and the peer.

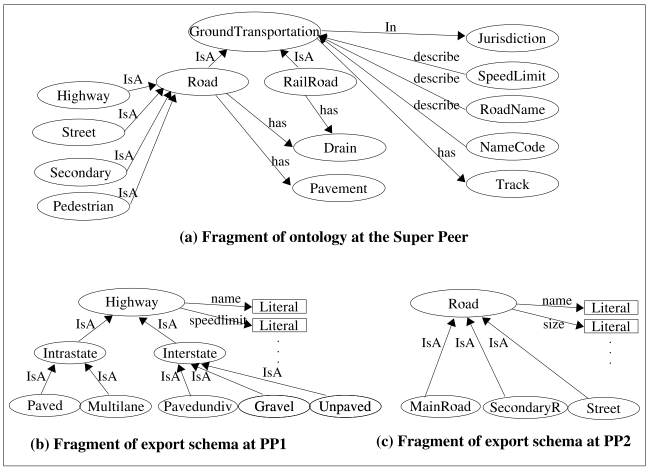


Fig. 2. Peer contents (fragments of ontology and export schema)

2.4 Discovery Resources

Discovery appropriate peers, which can respond a query is important issues. In our approach, the discovery processhas steps as follow: (1) a peer that requests information can utilize metadata information at super peer. The main purpose to get list of active peers. (2) Then, the peer as request sends his half agreement to the selected peers as sources. The sources peer will calculate *BindingValue* between their half agreements to half agreement of the peer. The calculation is based on similarity of concept. (3) Result of calculation matchmaking will be sent from sources to the peer. Refer to second step the request peer can ask to appropriate class and his properties. (4) After third step, appropriate sources peers have been selected, then request peers send a query to sources using mapping composition of two half agreement.

Result of half agreement between PP_n to SP can be utilized to make direct mapping between peers. The approach of mapping composition based on inverse mapping as follow: $\Omega_{ES_1 \rightarrow ES_2} = \Omega_1 * {\Omega_2}^{-1}$. Result of mapping composition is called full agreement unit.

3 Example

This example illustrates the steps for discovery sources using the general strategy of the agreement unit approach. Consider the peers PP1 and PP2 as providers and the fragments of ontology and export schemas shown in figure 2. Furthermore, assume that the two peers characterize roads differently. One peer PP1 classifies road according to speed limit while peer PP2 characterizes road according to size of road. Now consider a peer which characterizes roads by type (primary, secondary and so on) and which queries both peers PP1 and PP2 for a list of *secondary street in an area*. After developing of half agreement and look at meta information at super peer, the peer as request send his half agreement to PP1 and PP2. PP1 and PP2 calculate *BindingValue*. $BindingValue = \epsilon/\pi$, where ϵ = number of similar concept between request and provider peer, π = number of concept at request peer. PP2 will be selected as the interest parties because *BindingValue* of PP2 higher than threshold value (*BindingValue* of PP1=0/1, *BindingValue* of PP2=1/1). Result of discover can be continued to develop mapping composition and query process.

4 Conclusion

XMLS, RDFS and OWL and other ontology developments offer facility to enrich semantic description at P2P environment. We proposed a semantic agreement approach based on concept similarity values that take into account the place of a concept in a hierarchy and its structure consisting of directly linked properties and concepts. We utilize available approach of semantic mapping to develop half agreement. Result of half agreement can be utilized for discovery resources, mapping of concept between peers and handling of query to exchange information.

In the future work, we will focus on finalizing the architecture and prototype system to enhance negotiations between provider peer and a peer

References

1. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. The VLDB Journal **10** (2001) 334–250
2. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. J. Data Semantics IV **3730** (2005) 146–171
3. Kalfoglou, Y., Schorlemmer, M.: Ontology Mapping: The State of the Art. In: Semantic Interoperability and Integration. (2005)
4. Tempich, C., Staab, S., Wranik, A.: Remindin': semantic query routing in peer-to-peer networks based on social metaphors. In Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E., eds.: WWW, ACM (2004) 640–649
5. Siebes, R., Haase, P., Harmelen, F.v.: Expertise-Based Peer Selection. In: Semantic Web and Peer-to-Peer Decentralized Management and Exchange of Knowledge and Information. Springer (2006) 125–142
6. Lin, D.: An information-theoretic definition of similarity. In Shavlik, J.W., ed.: ICML, Morgan Kaufmann (1998) 296–304
7. Giunchiglia, F., Shvaiko, P.: Semantic Matching. Technical Report DIT-03-013, University of TN (2003)
8. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics **32**(1) (2006) 13–47
9. Euzenat, J., Valtech, P.: Similarity-Based Ontology Alignment in OWL-Lite. In: ECAI. (2004) 333–337
10. Bach, T.L., Dieng-Kuntz, R.: Measuring Similarity of Elements in OWL DL Ontologie. Technical Report ACACIA Project, INRIA Sophia Antipolis (2002)
11. Euzenat, J., Le Bach, T., Barasa, J., etc: D2.2.3: State of the art on ontology alignment. Technical Report IST-2004-507482, knowledgeweb (2004)

An Ontology Mapping Algorithm between Heterogeneous Product Classification Taxonomies

Wooju Kim[1], Sangun Park[2], Siri Bang[3], and Sunghwan Lee[4]

[1,3,4] Department of Information and Industrial Systems Engineering, College of Engineering,
Yonsei University, 134, Shin-Chon Dong, Seoul, South Korea
{wkim@yonsei.ac.kr,feeljazz@nate.com,siri27@nate.com}
[2] The u-City Research Institute, Yonsei University, 134, Shin-Chon Dong,
Seoul, South Korea {sangun.park@gmail.com}

Abstract. Research on ontology merging and mapping is one of the most important issues in the Semantic Web because ontologies are developed and used by various sites and organizations respectively. Electronic commerce is the area that require ontology mapping on product comparison over different product classification taxonomies of various shopping malls. But, a strict mapping strategy may lead a customer's configuration to search failure. Therefore we suggest a mapping algorithm for product matching that can provide more products by increasing sensitivity with reasonable decrease of specificity. We performed a comparative evaluation between our algorithm and PROMPT with 6 experimental sets.

Keywords: Semantic Web, Ontology Mapping, e-Commerce, Information Retrieval

1 Introduction

Research on ontology merging and mapping is one of the most important issues in the Semantic Web environment because ontologies are developed and used by various sites and organizations respectively. In electronic commerce area, each shopping mall has its own vocabulary and product hierarchy that cause a semantic interoperability problem [8]. Gathering and merging product information from tremendous shopping malls in most product comparison sites depends on manual work by human. But, it is extremely inefficient to manage promptly changing information about products. That is, electronics commerce is the domain which essentially needs automatic ontology mapping on product names and attributes for efficient product search over multiple shopping malls.

Most research on ontology mapping [1][3] focuses on precision because incorrect matching among different ontologies can cause severe problems. PROMPT [5] is one of the approaches that adopt such conservative strategies with exact matching. But, product search in comparison shopping requires more flexible mapping between user's configuration and products. According to the Boston Consulting Group [7], 48% of all users have experienced unsatisfactory search results on desired products and 28% of all product purchase tryouts could not reach purchase because of search failure. A strict mapping strategy that may involve search failure is not desirable because customers want rich information on products. Therefore, our research objective is to increase the number of matched products with the customer's configuration in automatic product mapping compared to the other ontology mapping approaches. This can be achieved by increasing recall rate with reasonable decrease in precision.

2 Sensitivity and Precision

Precision can be calculated by dividing the number of correctly matched terms by the number of all matched terms [2]. Therefore, if one wants to enhance precision, the best way is to minimize incorrectly matched terms. That is the reason that most approaches of ontology mapping adopt conservative and strict strategies. Meanwhile, **sensitivity** divides the number of correctly matched terms by the number of terms that should be matched [2]. Strict matching strategies try to increase precision as much as possible in spite of low sensitivity. But, those strict strategies are not desirable in comparison shopping as we mentioned in Section 1. **Specificity** is used with sensitivity together for classification performance measures and calculated by dividing the number of correctly not matched terms by the number of terms that should not be matched [2]. If we try to increase sensitivity by matching more products, specificity can be worse because correctly non-matched terms will decrease. Therefore, we use sensitivity and specificity in the performance evaluation and comparison of our algorithm and PROMPT.

Then, how to increase sensitivity compared to exact matching? The easiest way is using synonyms from WordNet [4]. By matching all synonyms of the given product, we can match more products and increase the chance of matching more correct products. But, it can also decrease precision. So, using synonym alone is not recommendable. In WordNet, a word has different senses and each sense has its own synonyms. If we can choose an appropriate sense of the given product from WordNet, it is possible to prevent precision from dropping too much by narrowing the synonym range. In this paper, we propose an ontology mapping algorithm for product matching based on above idea.

3 Product Matching through Ontology Mapping

3.1 Word Sense Disambiguation for Product Categories

Selection of an appropriate sense for a given product is important in order to keep precision at a reasonable level. If we use synonyms of all senses of the product, it will decrease precision because incorrect matching can increase. But, word sense disambiguation can enhance precision. The basic idea of word sense disambiguation is comparing a product hierarchy and hypernym hierarchies of senses of the product in WordNet. The sense *notebook* that is a computer has a different hypernym hierarchy with that of a book for notes as shown in Fig. 1. By comparing the product hierarchy of ODP (Open Directory Project) [6] in the left column of Fig. 1 and hypernym hierarchies in WordNet of the right column, we can choose a proper sense for *notebook*.

The first step of disambiguation is searching for hypernyms from a hierarchy of a sense that match with upper categories of the product as shown in the formula (1). *CS()* returns a set of hypernyms that match to a given upper category *x* from a given sense hierarchy *p*.

$$cs(x, p) = \{h \mid h \in SYNSETS(x) \text{ and } h \in hypernyms(p)\} \quad \text{where } x \text{ is an upper category of the product hierarchy} \tag{1}$$

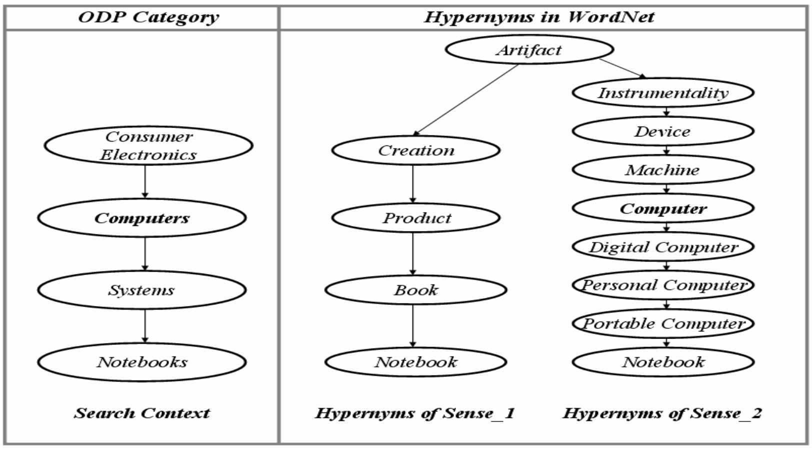


Fig. 1. A Product Hierarchy of ODP and Corresponding Hypernym Hierarchies in WordNet

The next step is calculating a measure represents the similarity between an upper category and a sense. If a matching hypernym is close to the sense, then the similarity is high because a closer hypernym is more important. The function *hypernymproximity()* returns the similarity by calculating a minimum distance between the matching hypernym and the *base* node of the sense in the hypernym hierarchy as shown in (2).

$$hypernymproximity(x, p) = \begin{cases} \dfrac{1}{Min\_dist(cs(x, p), base)} & if\ cs(x, p) \neq \phi \\ 0 & otherwise \end{cases} \tag{2}$$

The last step is calculating similarity between a product and senses. The function *pathproximity()* adds all *hyperproximity* of a given sense and divides it by the number of nodes of the product hierarchy as shown in (3).

$$pathproximity(p) = \frac{\sum_{x \in upper\_categories(base)} hypernymproximity(x, p)}{n} \tag{3}$$

3.2 Generation of Candidates for the Best Matching Category Path

Once we found an exact sense for the product from WordNet, the next step is to search for the candidates for the best matching category path from a target ontology. After the completion of search, we need to delete redundant categories of the product. To do this, the algorithm generates serial hierarchies of the categories by extracting all upper categories.

3.3 Choice of the Best Matching Product Category

To choose the best matching product category, we designed two measures for the calculation of similarities between the given product hierarchy and candidates. One is *co-occurrence* and the other is *order-consistency*. The measure, *co-occurrence* is the ratio of the number of common categories between a source hierarchy and a target hierarchy to the number of categories of the target hierarchy. However, *co-occurrence* is not enough to represent similarity because *co-occurrence* cannot measure orders of categories in the hierarchy. The other measure, *order-consistency* compares this order of categories. The final similarity between a source product and a target product is the average of *co-occurrence* and *order-consistency*. We choose a threshold on the similarity to determine whether we match the source product with the target product or not. We expect that the matching result will be changed by controlling not only the ratio of *co-occurrence* and *order-consistency* to the similarity but also the threshold.

4 Empirical Evaluation and Results

In this section, we compare the mapping results between our algorithm and PROMPT. PROMPT compares two different taxonomies and automatically recommends the matching terms by using synonyms [5].

To conduct an experiment, we selected two well-known shopping malls – Amazon.com and Buy.com – and ODP [6]. We constructed product ontologies from Amazon.com, Buy.com, and ODP respectively for our experiment. The product ontology of Amazon.com consists of 136 nodes, Buy.com consists of 225 nodes, and ODP consists of 133 nodes. A set of the experiment consists of one source ontology and one target ontology. Therefore, there are 6 sets in the experiment.

Table 1. Performance Results on Sensitivity and Specificity

Experimental Set	**Sensitivity**		**Specificity**	
	Our Algorithm	PROMPT	Our Algorithm	PROMPT
Amazon → Buy	96.9%	61.7%	56.4%	91.1%
Amazon → ODP	93.3%	25.7%	78.9%	84.5%
Buy → Amazon	93.5%	56.0%	61.0%	94.8%
Buy → ODP	97.2%	40.6%	69.5%	89.6%
ODP → Amazon	92.9%	36.0%	50.5%	88.1%
ODP → Buy	85.7%	60.9%	70.5%	84.7%
Average	**93.3%**	**46.8%**	**64.5%**	**88.8%**

Table 1 shows the performance results on sensitivity and specificity. On average, sensitivity of our algorithm is better than PROMPT by 46.5% and worse by 24.3%. It shows that our objective is successfully achieved. The maximum and minimum differences of sensitivity are 67.6% and 24.8% respectively while the maximum and minimum differences of specificity are -37.6% and -5.6% respectively.

5 Conclusion

In this paper, we proposed an ontology mapping algorithm that provides efficient product matching between heterogeneous product classifications. And we performed a comparative evaluation between our algorithm and PROMPT with 6 experimental sets. The experiment results showed that our algorithm is more effective than PROMPT in product comparison of the electronic commerce domain.

There is an interesting future research issue. Sensitivity and specificity can be changed by controlling not only the ratio of *co-occurrence* and *order-consistency* to the similarity but also the threshold as we described in Section 3. We expect that we can find the optimal values of the parameters – the ratio and the threshold. We are planning to conduct experiments finding the optimal values.

Acknowledgments. This research was funded mainly by the Ministry of Information and Communication in Republic of KOREA - National Project (Project management of Institute for Information Technology Advancement).

References

1. Ehrig, M., and Y. Sure: Ontology Mapping - An Integrated Approach. Lecture Notes in Computer Science, No. 3053 (2004) 76-91.
2. Han, J. and M. Kamber: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2000) 325-326.
3. Kalfoglou, Y. and M. Schorelmmer: Ontology mapping: the state of the art. The Knowledge Engineering Review, 18(1) (2003) 1-32.
4. Miller, G. A.: WordNet a Lexical Database for English. Communications of the ACM, 38(11) (1995) 39-41.
5. Noy, N.F., and M.A. Musen: The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. International Journal of Human-Computer Studies, 59(1) (2003) 983-1024.
6. Open Directory Project. http://www.dmoz.com (2006).
7. Pecaut, D., M. Silverstein, and P. Stanger: Winning the Online Consumer Insights into Online Consumer Behavior. A Report by the Boston Consulting Group, <http://www.bcg.com> (2000).
8. Veltman, K.H.: Syntactic and Semantic Interoperability: New Approaches to Knowledge and the Semantic Web. New Review of Information Networking, 7 (2001) 159-184.

Recognizing Emergent Nodes in Aligning Multiple Document Taxonomies

Tim Musgrove

TextDigger, Inc.
305 Vineyard Town Center #375
Morgan Hill, CA 95037 USA
tmusgrove@textdigger.com

Abstract. A document taxonomy alignment method, relying on document glosses and utilizing a soft ontology expansion, enables us to devise some all-new hierarchical leaf nodes for the purpose of better aligning a plurality of document taxonomies.

1. Introduction

In our past work of mapping different document taxonomies, we frequently were left with some "isolated nodes", i.e. categories of documents seeming to have no correlate in the other taxonomies. An example was in the Archery category on Yahoo, the sub-category of "Kyudo" (traditional Japanese archery). There was no equivalent to this category on DMOZ or About.com, the two taxonomies we were hoping to correlate. However, a soft ontology expansion we had devised to assist in the mapping meanwhile produced numerous candidate ontology nodes, such as "coaching/training" or "competitions/tournaments," and in this particular case, "traditional archery." While not a node in any of three reference taxonomies, "traditional archery" nonetheless applied to a great number of documents in all three, and especially in Yahoo's "Kyudo" category. Having used DMOZ as our "master taxonomy", we not only added "traditional archery" to it, but devised a method of automatically adding every other similar example, with the result of adding these new nodes: *Traditional Archery, Coaching & Training, Equipment & Gear, Stories & Discussion*.

The first of these, "traditional archery", included (as a child node) all the Kyudo documents, plus numerous documents from the other two indices, all of which pertain to traditional forms of archery. Since there are other traditional forms of archery (such as medieval European forms) besides Kyudo, it made sense that Kyudo be subsumed in the new node. We found that it was rather straightforward to devise a heuristic for automating this addition of nodes according to the following heuristic:

1. Find an expanded concept that is instantiated disproportionately in the document glosses of an unmapped node.
2. Test if that node is instantiated also in numerous documents not classified at a leaf node in a plurality of taxonomies.

3. If such a node is found, then create a new node with that concept and place the relevant documents under it.

In order to explain how this was accomplished, we will outline (1) our general approach to taxonomy node alignment by semantic resemblance; (2) our conception of a soft ontology expansion (3) the way in which results of the soft ontology expansion can be leveraged to create new nodes as described above.

2. Taxonomy alignment by semantic resemblance

One approach to taxonomy alignment is the intensional method, which examines the semantics of the names of the nodes, and the titles of documents, as well as the glosses applied to those documents by the taxonomy editors. We applied such a method to human-crafted document taxonomies bearing short glosses. These glosses are meant to summarize what the documents are about and what differentiates each one from others in the same topic, hence they are obviously valuable to our task.

We take the content words of the document titles and glosses, as well as bi-grams containing a topic word in any derived form (e.g., in the archery category we would take "field archery" and "archer's union", in addition to single words such as "arrows" and "bows"). We then check to see which of these may be closely related by semantic resemblance. For measuring semantic resemblance, we test for "semantic proximity" in WordNet, which we define as having a maximum distance of 2 in the WordNet hierarchy, with the additional limitations:

1. Only synonyms, hyponyms, hypernyms, and sister-terms are to be considered.

2. Sister-terms are considered proximate only if they share multiple content words in their glosses and/or example sentences in WordNet.

3. Hypernyms are included only if they are at least 4 levels down in the WordNet hierarchy from the root.

Note that this is similar to (Leacock 1998) in that it considers the depth of the taxonomy as counting toward semantic nearness, though our implementation is heuristic rather than statistical. (Since our application is to Web documents, we found it necessary to ignore certain words that are excessively frequent across all categories, and hence not useful, such as "photos", "contact details", "site map", etc.). Table 1 shows an outline of one of our case studies.

Table 1. Comparison of Archery in DMOZ, Yahoo and About.com

DMOZ	Yahoo	About.com
Chats & Forums	Bow Hunting	Shop for Archery & Bowhunting Gear
Clubs & Associations	Clubs & Organizations	Archery & Bowhunting Gear Manufacturers
Equipment Manufacturers	Competitions	Archery & Bowhunting Organizations
For Kids and Teens	Gear & Instruction	
Guides & Directories	Kyudo	
News & Media	Magazines	
Personal Pages	National Teams	
Tournaments & Events	Web Directories	

The result of our method is, for example, that "clubs" and "organizations" are treated as equivalent terms. This happens by means of a simple percentage match scoring of the content words in node names. For example, the pair of “Equipment Manufacturers” and “Archery and Bowhunting Gear Manufacturers” receives a score of 0.80, owing to the following facts: First, “Archery” is omitted because it is the same as the overarching topic of “Archery” and hence implicit in all node names. Second, the stop word “and” is discarded. Third, “gear” is matched to “equipment” as a hypernym. That leaves five words total, with only one of them (“bowhunting”) lacking a match: hence the score of 4/5 = 0.80. By trial and error we decided 0.66 was sufficient for alignment.

The virtue of this simple node name resemblance test is that it lets us align, for example, “Clubs and Organizations” with “Clubs and Associations” in two different taxonomies. However it leaves us with the different problem of the numerous documents not assigned a leaf node. In other words, in all three indices, many documents were simply classified in “Archery” without being assigned to a sub-category. In some cases, this seems correct, in that the documents in question were very general archery documents (or websites) not belonging to any particular sub-class. But in many other cases, it seemed that a node in a different taxonomy was a natural place for such documents. For example, a website of personal anecdotes and combined with feedback from others, was classified in one taxonomy simply as an “archery” document, but it would have found a perfect home in “Chats and Forums”. This defeats taxonomy alignment, in that it is implied that none of the documents in the one taxonomy would belong in “Chats & Forums” of the other – and yet many of them did.

This type of predicament was later resolved, in some cases, by the results of a soft ontology expansion of all three taxonomies. In other words, after having enriched the ontological characterization of each specific leaf node, we could often align it with an appropriate subset of the documents lumped together in a more general topic of a different taxonomy.

3. Soft ontology expansion of document taxonomy leaf nodes using WordNet

For this exercise, we went back to our extracted words and bi-grams (e.g. “calendar” and “field archery”, etc.), examined their WordNet glosses and example sentences and compared them with collocations and phrases in the document glosses, and found the following to hold true: if two words were frequently paired (collocated after skipping non-content words) in the taxonomy document glosses and also were found in each other’s WordNet glosses, they were, without exception (in our case studies), genuinely related and of ontological import in the category. Our operational definition of “frequent” was: having at least one occurrence in all three taxonomies and having multiple occurrences (2 or more) in at least two of three taxonomies.

This technique has similarities to (Beneventano 2003) and (Martin 2004), in that it employs WordNet to develop one’s taxonomy and/or ontology. The difference is that we are driving the process by reference to the glosses already created by editors of the various taxonomies. Our procedure derived the following soft concepts in Archery:

[calendar,schedule] having a relation to [event]

[tournament,competition] having relations to both [results] and [standings]
[outdoor] having a relation to [ranges]
[bow] having relations to [crossbow], [compound bow], and [long bow]

We call these "concepts" rather than merely "word occurrences" in view of the following: each is based on a small web of similar words, (e,g. "calendar"|"schedule"); each has an additional word relation ("events," etc.); all are contextualized to the local topic of Archery. The totality of all such extracted concepts we call a "soft ontology," in that it delineates raw materials of the local ontology, but obviously falls short of a formal representation of the relations between the concepts, such as those discussed in (Gaurino 1998).

Next, when checked the non-leaf-node documents' glosses for the presence of these concepts. If they matched, then we moved them to the newly created node. For example, several documents with glosses containing "discussion" and "stories" found their way into "Stories & Discussion." In the end, 37 of 189 documents were thus "migrated downward" to a leaf node, with the result that, on inspection, it seemed the alignment between taxonomies was more complete and intensionally unequivocal. This illustrates that taxonomy alignment cannot be divorced from issues of taxonomical scope and adequacy. If one taxonomy lacks the scope or granularity of another, then the only way to achieve proper alignment is to sort through some of the items in the less granular taxonomy so as to "multiply align" it to other nodes.

4. Emergent Nodes

Finally, we reached the result that certain of our soft ontology concepts embrace otherwise isolated nodes of one taxonomy, together with non-leaf-node documents of another. A clear example was the topic mentioned earlier, "Kyudo." Our soft ontology expansion had derived a "traditional archery" as a bi-gram. This was very dense in the Kyudo category (occurring in all but one of its items), while being found also in 16 non-leaf-node documents in DMOZ, including:

Donadoni **Archery**: Supplier of **traditional archery** equipment in Italy…
The **Archery** Centre: Specialists in field, **traditional**, and re-enactment **archery…**
Perris **Archery**: Recurve, compound and **traditional archery** equipment…

Our procedure was to use the concept string as a new node name (inserting "and" between words that had been found separately rather than directly collocated), and including as a child node the originally isolated node. So our master taxonomy now included "Archery/Traditional Archery/Kyudo" with several DMOZ documents placed in the new node "traditional archery." This satisfied us as being a far better alignment than we would have without the new node. Kyudo documents now had a closer parent than just being a direct child of "Archery." And the new interstitial node of "traditional archery" functions to explain where "Kyudo" belongs. We think the same of "Stories and Discussion" introduced as a parent of "Chat and Forums", and of "Coaching and Training" as a parent for "Instruction" documents that Yahoo had mixed with "Gear". Table 2 shows the overall alignment results.

Table 2. Results of alignment – New Nodes

New Nodes -Child node	DMOZ	Yahoo	About
Stories & Discussion - Chats & Forums	Chats & Forums	Glosses with "stories, ""discussion"	Glosses with "stories, ""discussion"
Equipment and Gear	Equipment Manufacturers	Glosses with "equipment" and "gear"	Archery & Bowhunting Gear Manufacturers, Shop for Archery & Bowhunting Gear
Bow Hunting	Glosses with "bow hunting"	Bow Hunting	Glosses with "bow hunting"
Coaching & Training	Glosses with "instruct", "coach", "train"	Glosses with "instruct", "coach", "train"	Glosses with "instruct", "coach", "train"
Traditional Archery - Kyudo	Glosses with "traditional"	Kyudo	Glosses with "traditional"

Regarding accuracy, the introduction of new nodes carried just one misclassified document, which had been misclassified already on of the third party indices. In general, the accuracy of this method should be as good as the accuracy of the classification of the participant taxonomies. However, we are concerned about the naming of the newly created nodes. In the Archery case above, all the names read nicely, but when we did Soccer, one node received the name "Instructing" when we would prefer to see "Instruction." Further work could be done on node naming

5. Conclusions

Editorially created document glosses are a boon to taxonomy alignment, in that they constitute a rich resource to guide semantic resemblance analysis, and have the added bonus, when soft ontology expansion is applied via WordNet, of enabling us to create new interstitial nodes for a more complete and unequivocal alignment of taxonomies.

References

Beneventano D., Bergamaschi S., Guerra, F., Vincini, M. 2003. Building an integrated Ontology within SEWASIE system. *Proceedings of the First International Workshop on Semantic Web and Data-bases (SWDB)*

Gaurino, N. 1998. Formal ontologies and information systems. *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS'98)*, Trento, Italy.

Leacock, Claudia, Martin Chodorow & George A. Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics,* 24(1): 147-165.

Martin, T. ., Ben Azvine, Behrad Assadian. 2004. Acquisition of Soft Taxonomies. *IPMU-04*, 1021-1032.

FORPM: Boosting Users' Effect on Ontology Matching

Dunwei Wen[1,2], Xiaohu Fan[1], Fuhua Lin[2]

[1] School of Information Science and Engineering,
Central South University, Changsha, Hunan 410083, China
`dwwen@mail.csu.edu.cn, fanxiaohu17@gmail.com`
[2] School of Computing and Information Systems,
Athabasca University, Athabasca, Alberta T9S 3A3, Canada
`{dunweiw, oscarl}@athabascau.ca`

Abstract. In this paper, we attempt to view the ontology matching task from an information gaining angle. In our opinions, the information used for matching mainly comes from the matching tools as well as the human experts. With this understanding, we believe that by making good use of user efforts, we can also accelerate the matching process. Hence we present a prototype system named FORPM. First, it ranks the entities of the ontology. Important entities are chosen as centroids to form fragments. Then, users can use those centroids' information to estimate the content of the fragments and initially match them. Finally, automatic matching is carried out among those matched fragments. Experiment results obtained so far show that with a few user efforts, our approach significantly improves the matching efficiency while the loss of accuracy is acceptable.

1 Motivation

Ontology matching aims at finding semantic relationships between entities of different ontologies for solving the interoperation problem. From the viewpoint of information theory, we view a matching problem Δ as a process of information gaining with uncertainty $\Omega(\Delta)$. Let φ denotes the information obtained by the matching tool (from ontology itself as well as external source like WordNet), ω denotes the information provided by users in the validation step (we hold the same kind of opinions with [1] that fully automatic ontology matching is still impossible). To obtain matching results of high quality, we believe that the following equation has to be satisfied:

$$\omega + \varphi \geq \Omega(\Delta). \tag{1}$$

On one hand, to our best knowledge, recently numerous researches have been focused on how to maximize φ and made great progresses. On the other hand, we believe that the human users, especially the domain experts are capable of discovering complex relationships, such as more general ($\supseteq$), less general ($\subseteq$) etc., between candidate pairs. This extra information, however, is always ignored and human effort is simply used in the validation step to judge simple relations such as matching or not matching. With these understandings, our work is aiming at making good use of the information provided by users, thus accelerating the matching process.

In this paper we propose: 1) an information theory based model for concept ranking and centroid extraction; 2) a clustering algorithm for ontology partitioning. To test our approch, we also introduce a prototype system called FORPM.

2 FORPM (Framework for Ontology Ranking, Partitioning and Matching)

FORPM is implemented with Java under JDK 5.0 and Eclipse 3.1.2. The system architecture is shown below in Fig 1. First, two ontologies are input and then transformed into DAG (Directed Acyclic Graph), where the "is-a" relations are transformed into arcs and concepts are transformed into nodes. After the four main process steps in the dash line, the result and a reference-mapping file are sent to the evaluation module, in which the evaluation results are generated automatically and presented to the user.

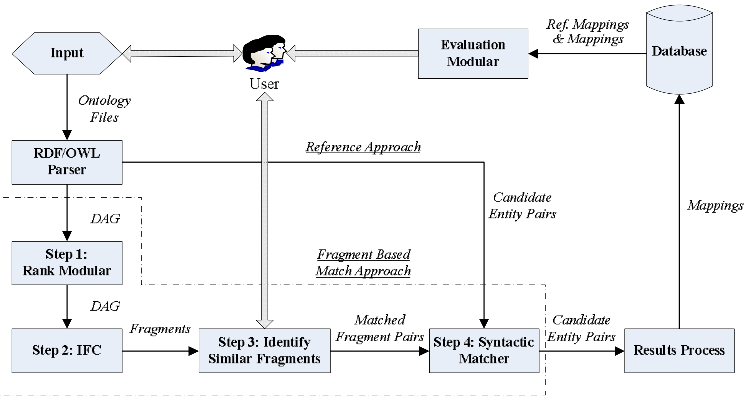


Fig. 1. System Overview

Step 1: Entity Ranking Based on our observation, the amount of information provided by a user equals the sum of the amount of information I_i provided in T times of the user's validation.

$$\omega = \sum_{\{i|i\in[1,T],i\in\mathrm{N}\}} I_i. \tag{2}$$

If we assume that the cost of the user's every validation be the same, then one intuitive way to improve the matching efficiency is to maximize I_i wisely. Hence fundamental to our ranking approach is the ability to measure how much information is conveyed in a node thereby giving a sense of how much information the computer would "gain" by being informed about a discovered matching pair.

In information theory [2], the amount of information contained in an event is measured by the negative logarithm of the probability of occurrence of the event. Thus if χ is an event that has possible outcome values $x_1, x_2, ..., x_n$ occurring with probabilities $pr_1, pr_2, ..., pr_n$, the amount of information gained or uncertainty removed by knowing that has the outcome x_i is given by:

$$I(\chi = x_i) = -\log(pr_i). \tag{3}$$

Based on this we can build a model to measure the amount of information of a node in an ontology graph by considering the concept as an event and "is-a" relations as its outcomes. Assume that in the ontology graph $G(Arcs, Nodes)$, where $Arcs$ is the set of all "is-a" relations and $Nodes$ is the set of all concepts in ontology O. Then for any arc $a_i \in Arcs$, its probability is given by

$$\Pr(\chi = a_i) = \left(\frac{1}{|arcs_i|}\right). \tag{4}$$

Here $|arcs_i|$ is the number of arcs connecting with Node n_i. Thus the amount of information contained in arc a_i is:

$$I(\chi = a_i) = -\log(\chi = a_i). \tag{5}$$

If we have a node $n_i \in Nodes$, then the amount of information contained in n_i is:

$$I(n_i) = \sum_{a_i \in arcs_i} I(a_i). \tag{6}$$

We use the amount of information to rank nodes. The node contains the most amount of information is defined as an information center.

Definition 1 (Information Center/Centroid Node). *Let G(N, A) be an ontology graph, N be the node set, A be the arcs set, then node $I_c \in N$ is an information center if for any node $n_i \in N$:*

$$I(I_c) \geq I(n_i). \tag{7}$$

Step 2: IFC (Information Flooding theory based Clustering)

The goal of this step is to form fragments from centroids. We noticed that in the "is-a" hierarchy tree, semantic similarity between two concepts often decays as the distance between them increases. In our work, we define an information flooding function to measure how strong a source node could affect a target node.

Definition 2 (Information Flood). *Let G (N, A) be an ontology graph, n_i, n_j ∈N, we define the information flood from n_i to n_j as:*

$$InfoFlood\left(n_i, n_j\right) = F\left(Dist_{ij}\right) I(n_i). \tag{8}$$

Where $I(n_i)$ is the information contained in n_i, $F(Dist_{ij})$ is a quadratic experiential decay function to simulate the attenuation of similarity defined as

$$F(Dist_{ij}) = \frac{1}{a \times Dist_{ij}^2 + b \times Dist_{ij} + c}. \tag{9}$$

and $Dist_{ij}$ is the number of arcs between node n_j and node n_j in the "is-a" relation hierarchy tree. In our experiment, we have $a = 0.25$; $b = 0.5$; $c = 0$.

Definition 3 (Fragment). *Let O be an Ontology, $G_i(N, A)$ be a graph representing part of O, where N is the node set, A is the arc set. Let dList be the set of all centroid nodes in O. If for any $d_j \in dList$ and all $n_k \in N$, we have $d_i \in dList$ which satisfies*

$$InfoFlood(d_i, n_k) \geq Max\left(InfoFlood\left(d_j, n_k\right)\right). \tag{10}$$

Then we say G_i is a fragment $f(d_i,m_i)$ with m_i as its size and d_i as its centroid node.

Table 1. IFC (Information Flooding theory based Clustering)

```
Procedure (G, Max, Min)
  for each node in Graph G
     Ranking nodes, the first Max nodes with highest rank are centroids;
  end
  while the centroid list does not change or iteration times < SetValue
     Compute the InfoFlood($n_i$,$n_j$) from centroids to other normal nodes;
     Assign normal nodes to centroids which maximize the InfoFlood(ni,nj);
     Form the fragments set F;
     Recalculate the centroids list;
  end
end
```

We briefly describe the partitioning algorithm in Table 1. The algorithm receives two parameters, Max, the upper bound of the number of the fragment, and Min, the lower bound of the size of the fragment. We set a max iteration number to ensure the stop of the algorithm. Also a merge algorithm is implemented to deal with the fragments whose size is below the lower bound.

Step 3: Manual Matching.

In this step, users use those centroid nodes to estimate the content of the fragments and then match them manually. A centroid node may have more than one counterparts with relations such as equivalence (=), more general($\supseteq$), less general($\subseteq$), mismatch ($\perp$) and overlapping ($\cap$). Two centroids are considered semantically matched [3] if the relation between them is not mismatch.

Step 4: Automatic Matching.

Two fragments are viewed as matched if their centroids are semantically matched, the remaining matching work between two matched fragments is the same as a normal task but of smaller sizes. Various approaches could be adopted here to finish the task. In FORPM, we employ the same string-based tech in [4] for demonstration.

3 Evaluation and Discussion

In our experiment, we adopted a dataset from [4]. The Russia1a contains 151 concepts while Russia1b contains 162 concepts, with 64 human confirmed mappings (concepts only). We used F-Measure [cf.4] and Cost (see blow) as quility metrics.

$$Cost = \frac{\#Compare\_Times}{\#Found\_Marched\_Pairs}. \tag{11}$$

In FORPM, users can tune the system by changing the value of Max and Min in step2. We can see in Fig. 2 that the more fragments there are, the more likely users are to make right judgememts which lower the cost. Meanwhile more human work is required (The user has to do Max*Max times validations at most). According to our experience, it seems that the program performs best when the parameter Max is set between 5% ~ 10% of the total number of the concepts.

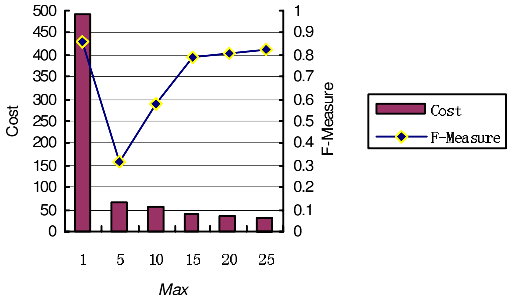


Fig. 2. FORPM with different Max (Min = 4)

4 Concluding Remarks

In this paper, we have proposed an information gaining theory based framework for ontology matching. We have shown that with a few user efforts, our approach is effective in reducing the matching complexity.

Our work is inspired by data mining technology. We gain our idea of information model from [5]. Our tool refers to [6]'s work in implementation, while [7] propose an automatic block based matching approach. Both [7] and our tool employ a ranking step to label the blocks. However, [7]'s rank step is after the partitioning step, while in FORPM, ranking step is firstly carried out since we employ an extraction-like clustering algorithm.

5 Acknowledgements

We thank National Science and Engineering Research Council of Canada, Academic Research Council of Athabasca University for their financial support for the research, and thank anonymous reviewers for their detailed and constructive comments.

References

1. N.F.Noy and M.A.Musen: The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*.2003,59(6):983-1024.
2. C.E.SHANNON: A Mathematical Theory of Communication. *The Bell System Technical Journal*.Vol. 27, pp. 379-423,623-656,July, October,1948.
3. Paolo Avesani, Fausto Giunchiglia, and Mikalai Yatskevich: A Large Scale Taxonomy Mapping Evaluation. *ISWC 2005*.LNCS 3729, pp. 67-81, 2005.
4. Marc Ehrig and Steffen Staab: QOM- Quick Ontology Mapping. *The Semantic Web Proceedings ISWC'04*.pp. 289-303, 2004. Springer-Verlag LNCS 3298.
5. Kemafor Anyanwu Angela Maduko and Amit Sheth : SemRank: Ranking Complex Relationship Search Results on the Semantic Web In: *WWW 2005* May 10-14, 2005, Chiba, Japan.
6. Heiner Stuckenschmidt and Michel Klein: Structure-Based Partitioning of Large Concept Hierarchies. In: *The SemanticWeb Proceedings ISWC'04* pp. 289-303, 2004.
7. W. Hu, Y. Zhao, Y. Qu: Partition-Based Block Matching of Large Class Hierarchies In *Proceedings of ASWC 2006*, Beijing, China, 2006.

OntoMas: a Tutoring System dedicated to Ontology Matching

Mirella Huza[1], Mounira Harzallah[1], Francky Trichet[1]

[1]LINA – University of Nantes
Laboratoire d'Informatique de Nantes Atlantique (FRE-CNRS 2729)
Team « Knowledge and Decision »
2, rue de la houssinière – BP 92208 - 44322 Nantes, France
{mirella.huza,mounira.harzallah,francky.trichet}@univ-nantes.fr

Abstract. Ontology matching is now a core question in most of the applications that require semantic interoperability. To deal with this problem, a lot of methods, classified according to different criteria, are currently developed. However, choosing the most relevant method in a particular context is not an easy task since it requires to know all the methods and their intrinsic properties. The objective of the OntoMas[1] tutoring system (*Ontology Matching Assistant*) is (1) to propose an architecture and to develop an effective knowledge-based system dedicated to a fine-grained description and a classification of the current matching methods and (2) to provide functionalities dedicated to the definition of advices and explanations (for the end-user), in order to facilitate both the choice of the most suitable method for a particular matching problem and the learning of this new domain: ontology matching.

Keywords: Ontology matching, Classification of ontology matching method, Tutoring System, Ontology matching assistant.

1. Introduction

Ontology matching is now a core question in most of the applications that require semantic interoperability such as the Semantic Web. To deal with this problem, which mainly consists in finding semantic links (*e.g.* equivalence, disjointness or subsumption) between the concepts and the relations of two ontologies covering overlapping domains, a lot of methods are currently proposed [1,2,3,4].

The choice of a method or the combination of several ones is not an easy task since (1) it depends on multiple criteria related both (i) to the characteristics of the current matching methods, (ii) to the ontologies that are considered for the matching process (in terms of type, structure, representation format, etc.) and (iii) to the end-user preferences and (2) it requires to know all these different criteria in order to select the best method for the considered matching context. Moreover, as the

---

[1] This work is supported by the Commission of the European Communities under the Sixth Framework Program: [INTEROP Network Of Excellence – 508011] http://www.interop-noe.org.

Ontology Matching domain is currently booming, it is really difficult to keep an up-to-date overview of the current methods.

This paper introduces the basic foundations of a Tutoring System dedicated to Ontology Matching. This system, called OntoMas[2] (*Ontology Matching Assistant*), aims at facilitating the classification of ontology matching methods and at helping the end-user to select the most relevant method(s) according to his matching problem. It also aims at providing to the end-user relevant advices and explanations on the method(s) that can be used (or can definitively not be used) in his matching context. Thus, OntoMas is not only a tool to select the best matching technique; it is also a tool dedicated to ontology matching learning.

2. OntoMas matching method classification

In most of the current classifications, only the criteria which directly relate to the description of the methods are considered. However, the choice of a matching method also depends on the end-user matching context. Since our objective is to propose a new, complete, up-to-date and easy to use classification of both the current matching methods and the possible matching contexts, we propose to differentiate two categories of classifications (and thus two groups of criteria): a classification based on the characteristics of the matching methods and a classification based on the characteristics of the matching context.

2.1 Matching method-based criteria

For the matching method-based criteria, we reuse the three dimensions of [4] and enrich them with new criteria.

For the *Input dimension*, the criterion **Interlingual matching** is added: is the method able to perform interlingual matching between ontologies, *e.g.* matching an English and a French ontology ? We add also a sub-dimension including the following criteria: (i) **Internal representation** (the method can translate the input models into an internal representation of ontology specification language), (ii) **String normalization on input data** (the method can perform string normalization on the input ontologies, *e.g.* case normalization, diacritics suppression, etc.).

For the *Processing dimension*, three criteria are introduced. The first one is about **User interaction**. The user interaction can be used in different matching phases: preprocessing, configuration of parameters, definition of an initial set of matches; modification of the architecture of the combined matchers and validation of results [2]. The second one is the **Iteration** criterion. The method can offer the possibility of performing iterations in the matching process, *e.g.* the matching algorithm can be launched several times, in this way the quality of the results can rise considerably. The third criterion, called **Alignment extraction**, refers to the process of choosing from all the obtained matching a satisfactory set of correspondences between

---

[2] The current prototype of OntoMas is available at the following URL: http://www.polytech.univ-nantes.fr/ontomas/

ontologies. It can be achieved by: (i) displaying the entity pairs with their similarity scores and leaving the choice of the appropriate pairs up to the user, (ii) threshold-based filtering, selecting correspondences over a particular threshold, thus retaining only the most similar entity pairs [2].

For the *Output dimension*, we add three others criteria introduced in [2]: (i) **Matches format** (the format in which the matching is represented - it can be: matching pairs, OWL, XML, ASCII, SKOS, text or other format); (ii) **Complete matching** (the method can produce complete matching for one of the ontologies); (iii) **Injective matching** (the method can produce injective, and reversible matching); (iv) **Integration** (using the results of ontology matching for ontology merging).

Finally a G*eneral dimension* concerning the tool description is introduced. Its relevant criteria are: (i) Type of solution (method or tool). When the solution is a tool: (ii) **Programming language** (it was implemented using a programming language); (iii) **Free version** (it can have a free/not free version); (iv) **Installation requirements** (it may have installation requirements: need installation of another software).

All these criteria are method-oriented. Thus, they are more dedicated to the ontology matching engineer point of view.

2.2 Matching context based criteria

The second group of criteria are dedicated to the description of the matching context: (1) *Input ontologies criteria* , (2) *User desired output criteria* and (3) *Method desired characteristics criteria*. These criteria are also related to the matching process, but from the end-user point of view.

The *Input ontologies criteria* are dedicated to the description of the content of the ontologies that are considered: (i) Format; (ii) Language (the natural language used in the representation of ontology); (iii) Content; (iv) **Ontology Domain** (*e.*g. Human Anatomy, Computer Science); (v) **Ontology size** (ontology is very large or not, A very large ontology is an ontology that has more than 1000 concepts); (vi) **Ontology structure** (it concerns the unbalance between the ontologies, one is a deep ontology and the other one is a shallow ontology), (vii) **Ontology version** (the ontologies are different versions of the same ontology), (viii) **Ontology similarity** (the domains covering by the ontologies are very different – they are not strong overlapping domains), (ix) **Ontology depth**, (x) **Ontology Relations** (*e.*g. is-a, part-of). The three first criteria are already introduced in the classifications presented in Section 2.

For the criteria related to the *User desired output*, we propose to consider: (i) Matches cardinality, (ii) Matches format, (iii) Matches Relations, (iv) Matches Ranking, (v) Integration.

The *Method desired characteristics criteria* are: (i) Type of solution (method or tool), (ii) Programming language, (iii) Free version, (iv) Iteration and (v) User Interaction.

3. OntoMas Tutoring System

One of the functionalities of OntoMas is the selection of the most suitable matching methods (for matching two ontologies given by the end-user) from the ones integrated in the knowledge base which currently includes 3 methods (IF-Map - Information Flow based ontology mapping, HCONE-merge and AROMA - Association Rule Ontology Mapping Approach) and 5 tools (CMS - CROSI Mapping System, AUTOMS, TooCoM, Organon and H-MATCH). The principle underlying this selection consists in using a set of **decision rules** defined between (1) methods described in OntoMas knowledge base and (2) matching contexts.

A **decision rule** represents a relation between a criterion related to the method point of view and a criterion related to the context point of view. The selection process is based on all the decision rules of the OntoMas system.

Each characteristic of the matching context, after being described by the end-user, determines a "restriction" imposed on the methods that can be used in the matching context. Each characteristic of the matching context implies a decision rule. Thus, we use 20 rules: 10 for the *Input ontologies criteria*, 5 for the *Method desired characteristics criteria* and 5 for the *User desired output criteria*.

Three types of decision rules are distinguished: compulsory rules, not compulsory rules and optional rules.

(C) Compulsory rules. These rules are related to the criteria used to describe the matching context for which the end-user has to provide values because they correspond to the minimum amount of information which is necessary to perform the decision process.

Example of C: Format

If	Ontology 1 is represented in the format F1 And Ontology2 is represented in The format F2 And F1 and F2 are the same or compatible
Then	SELECT from the knowledge base the methods that can use as input models ontologies that are represented in the F1(F2) format.

(NC) Not compulsory rules. They correspond to the criteria for which the end-user is not obliged to provide (input) information. Example of such a rule: Size.

Example of NC: Size

If	Ontology1 is a very large ontology Or Ontology2 is a very large ontology
Then	Increase with one point the total score of the methods that produce relevant results when the input ontologies are very large.

(O) Optional rules. They allow the end-user to precise preferences ("Only find methods that…" OR "I prefer methods… that"). When the end-user selects the "Only find" option, then the rule becomes compulsory and when the end-user selects the "I prefer" option, the rule becomes not compulsory.

Exemple of O: User interaction

Compulsory

If	The end-user wants to CHOOSE ONLY methods that use (don't use) Interaction
Then	Choose from the knowledge base the methods that use (don't use) in their matching phases user interaction.

Not compulsory

If	The end-user wants OntoMas to PREFER methods that use (don't use) Interaction
Then	Increase with one point the total score of the methods that use (don't use) in their matching phases user interaction.

After applying all the compulsory rules to the methods of the knowledge base, a set of methods that satisfy the compulsory requirements of the matching context is obtained: setC. Then, each rule of the not compulsory rules is applied to each method m_i of setC. If m_i satisfies the rule, one adds to its importance score one point. If m_i does not satisfy the rule, one decreases its importance score with one point. After applying all the not compulsory rules to the setC, an importance score is obtained for each method from the setC. It is then possible to order the set (more higher the importance score is, more suitable the method is).

5 Conclusion

In this paper, we have underlined the fact that choosing the best method for a particular ontology matching problem is not an easy task since the number of methods is currently booming and the criteria used in the current classifications do not enable the description of the matching context (they are mainly method-oriented). To deal with this problem, we propose to introduce new criteria dedicated to the end-user point of view: (i) description of his ontologies (domain, size, structure, version, similarity, depth, relations) and (ii) description of his preferences in terms of results (matches cardinality, format, relations, ranking and integration) and tools (programming language, free version, iteration and user interaction). Based on these new criteria and on the basic method-oriented criteria, we have defined compulsory, not compulsory and optional decision rules. These rules are used by OntoMas in order to recommend (and order) a set suitable (resp. unsuitable) methods for a specific ontology matching problem.

References

1. A. Doan and A. Halevy. Semantic integration research in the database community: A brief survey. 2005.
2. J. Euzenat, J. Barrasa, P. Bouquet, R. Dieng, M. Ehrig, M. Hauswirth, M. Jarrar, R. Lara,D. Maynard, A. Napoli, G. Stamou, H. Stuckenschmidt, P. Shvaiko, S. Tessaris, S. van Acker, I. Zaihrayeu, and T. L. Bach. D2.2.3: State of the art on ontology alignment. Technical report, NoE Knowledge Web project deliverable, 2004.
3. N. F. Noy. Semantic integration: A survey of ontology-based approaches. SIGMOD Record, 33(4):65–70, 2004.
4. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. 3730:146–171, 2005.

SOMET: Shared Ontology Matching Environment

Liam Magee

Globalism Institute, School of Global Studies, Social Sciences and Planning
RMIT, Melbourne, Australia
`liam@commonground.com.au`

`http://www.rmit.edu.au`

Abstract. In this paper we present a tool, SOMET, for collaborative developing, matching and merging ontologies. The tool's design is based on a Wiki model, allowing for multiple authors to contribute to an ontology. It also provides a number of meta-ontology features, including the ability to compare, match and merge. The tool makes use of one algorithmic approach to element-level mapping, demonstrating the use of both automated and manual matching.

1 Introduction

Much research has been invested into automated techniques of ontology matching [6, 7]. There is general recognition of the need to augment these techniques with manual matching. *Collaborative* matching, utilising both automated and manual matches, is important to resolve conceptual ambiguities, and to promote re-use across organisational and geographical boundaries.

This paper presents an online tool, SOMET (pronounced 'Summit'), for collaborative developing, matching and merging ontologies. It is motivated by the idea that ontology development and matching is essentially a *social* and *interactive* activity, and is best served by tools which permit this. As such, the tool is based on the Wiki model for ontology development.

This paper is structured as follows: In Section 2, we review related work. Section 3 presents a case for collaborative ontology development and matching. Section 4 describes the design and implementation of the SOMET prototype, along with our approach to ontology matching and merging. Section 5 examines the test results of using SOMET on two sample ontologies. In Section 6, we look at further directions for SOMET. Section 7 concludes the paper.

2 Related Work

The concept of shared or collaborative ontology development is not new. A number of tools have been introduced, including CODE [5], KAON [3], OntoEdit [9], Ontolingua [4], WebODE [1] and Wiki@nt [2]. SOMET differs from these tools in focussing particularly on ontology *matching* and *merging* in a collaborative environment.

There has been considerable exploration of approaches for ontology matching. Much of the research has been into finding suitable algorithms to automate part or all of a given matching task. As shown by surveys [6, 7], such algorithms use a variety of strategies for matching ontologies. This paper explores the use and partial implementation of one such algorithm, S-MATCH [8]. The S-MATCH algorithm emphasises its semantic matching characteristics. However the algorithm also exploits syntactic and external techniques. As such, it is a good candidate for exploring the use of semi-automated techiques in a collaborative environment.

Recent work also suggests the importance of community-driven ontology matching [10]. This paper assumes development and matching of ontologies is often collaborative in nature, and requires tools such as SOMET to realise the benefit of community-driven domain models.

3 Overview of Collaborative Ontology Development and Matching

Ontologies are typically developed by a process of iterative *construction* and *consultation*, with a focus on concepts in a specified domain. For the most part construction and consultation are separate activities, conducted in serial fashion, as the modeller uses specific knowledge of the modelling environment to apply the results of the consultation process. In the case of traditional database and software engineering activities, such established practices are generally entrenched in broader lifecycle processes, and there has been little impetus to shift the onus of model construction from the modelling expert. For Semantic Web ontologies, where models are frequently shared across organisational and geographical boundaries, there is significantly greater motivation to develop such models collaboratively.

The approach used in this paper is based on the success of open access content systems. The approach accepts that a lower grade user interface will be acceptable in certain contexts, just as authoring content online is an acceptable degradation from using a word processor in certain contexts. In particular, if the ontology is small, then the frequent submission and retrieval of ontology elements across a network of ontology components will be tolerable.

4 Outline of SOMET System, and Matching and Merging Techniques

SOMET is a prototype that has been developed in Ruby, using the Ruby on Rails framework. It employs a commonly used model-view-controller architecture. It also employs a plug-in architecture for executing matching algorithms.

The SOMET interface makes it possible to construct an ontology with classes, properties and individuals. Notable features at this stage include the following:

- *Creation* and *editing* of ontologies, classes, properties, individuals and annotations.
- *Importing* and *exporting* an ontology.
- Generation of a *comparative report* of differences between two ontologies.
- Manual and semi-automated *matching* of classes.
- *Merging* of two ontologies.
- Various Wiki features, such as *publishing*, *sharing*, *publishing*, *versioning*, *logging* and *commenting* on ontologies. Class and property matches can be proposed, discussed, and approved or rejected.

We have conducted a partial implementation of the S-MATCH algorithm [8]. As there is not a suitable satisfiability engine in the Ruby language, we have not been able to implement Step 4 of the algorithm. Consequently we have not yet been able to test the *semantic* aspects of S-MATCH, which require the translation of the labels of the path of a given node on the ontology graph into a propositional logic formula. Instead we have translated steps 1-3 of S-MATCH to Ruby, using the OWL object model we have developed. We were able to develop a matrix of ontology labels with at least partial asssignments of the following relations: *equivalence* ($=$), *more general* ($\sqsupseteq$), *less general* ($\sqsubseteq$), and *disjointness* ($\perp$). The result is an element-based, *syntactic* and *external* technique, as outlined in [**?**] - but without the key *semantic* characteristics outlined in [8]. The following outlines the key steps of the algorithm, as presented in [8, 305-7]:

Step 1. For all labels L in the two trees, compute *concepts* of *labels.*

Step 2. For all nodes N in the two trees, compute *concepts* of *nodes.*

Step 3. For all pairs of labels in the two trees, compute *relations* among *concepts* of *labels.*

Step 4. For all pairs of nodes in the two trees, compute *relations* among *concepts* of *nodes.*

For our purposes, we utilised WordNet in steps 1 (to develop the senses of each lemma) and 3 (to generate the relations between pairs of labels). The generation of the label matrix at step 3 made use of an exhaustive traversal through the WordNet dictionary. The results of step 3 are a set of tuples, $\langle eID, n1\_i, n2\_j, R\rangle$, where eID is a unique identifier for the given element-level match, $n1_i$ is the i-th node of the first ontology graph, $n2_j$ is the j-th node of the second ontology graph, and R is one of equivalence, more general, less general or disjoint. We ignore possible overlapping or unknown relations. These tuples are captured in a database, grouped together as an ontology match.

The *Merge* operation generates a new ontology graph from two existing ones, on the basis of a defined ontology match. The resulting merged ontology is stored in the database. The operation has 5 steps:

1. Compute the set of direct *child-parent* relations for the new graph, based on the set of element-level matches.

2. Compute the set of direct *parent-child* relations for the new graph, based on the results of step 1.
3. Generate the complete set of parent and child nodes for the new graph, based both on relations from steps 1 and 2, as well as the existing relations in the source graphs.
4. Compute the set of disjoint relations for the new graph, based on the set of element-level matches, for all siblings in the graph generated by step 3, where such siblings are not already disjoint.
5. Perform a deep copy of non-matched objects from each of the source ontologies into the new ontology.

5 Test Results

The matching and merging capabilities of SOMET have been tested using two simple ontologies. The goals of the test were to successfully invoke the S-MATCH implementation, generate a set of class-level matches, add or modify one such match, and to merge the two ontologies into a third on the basis of the matches. The test would be successful a) if the merged ontology contained the union of the two source ontology classes in a directed acyclic graph, with at least some successful matches, and b) if the *Match* and *Merge* operations execute in reasonable time. Tests were conducted on a P4 3.0GHz machine with 2GB of memory.

The matching operation took 74.407 seconds. Further analysis showed the majority of this time was due to the exhaustive scansion of the WordNet database. The merging operation took 5.703 seconds. The results show some inconsistencies of the WordNet associations, with certain anomalous subsumtion relations identified.

The results of the test show that for small ontologies, the *Match* and *Merge* operations can be conducted in an shared online environment. While the performance is sub-optimal, this could be corrected by depth-limited WordNet searches, and augmented with domain-specific vocabularies. The quality of the match also varies, although implementing step 4 of the S-MATCH algorithm, ie. the satisfiability checks on formulas representing the full path of any given element, would improve this greatly. The test also demonstrates the ability to augment automated matching techniques with manual matching.

6 Further Work

SOMET has been developed to a prototype level, and as such lacks the kinds of user interface enhancements expected of such a tool in a production environment. To realise the aims of providing a generic matching tool with ‘pluggable’ matching techniques, a more sophisticated plug-in architecture needs to be developed. Given the diversity of matching approaches, the viability of this aim also needs to be better ascertained.

7 Conclusions

Research on publishing and document standards over several years has motivated the investigation into how ontologies can be developed and matched in a collaborative way. We have concluded *social interaction*, *negotiation* and *collaboration* are necessary aspects to successful ontology matching in many environments. So far, tools for ontology matching have focussed on *private* ontology matching. In this paper we have presented SOMET as a prototype for *collaborative* ontology development, matching and merging. The test results have been encouraging in terms of its utility for these purposes.

Acknowledgments. This work has been supported by ARC Grant LP0667834, 'Towards the Semantic Web: Standards and Interoperability across Document Management and Publishing Supply Chains'.

References

1. J. C. Arpirez, O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez. Webode: a scalable workbench for ontological engineering. In *Proceedings of the international conference on Knowledge capture*, pages 6-13. ACM Press, 2001.
2. A. Bao, V. Honavar. Collaborative Ontology Building with Wiki@nt - A Multi-agent Based Ontology Building Environment. In *Proceedings of the 3rd International Workshop on Evaluation of Ontology-based Tools (ISWC 2004)*. Technical Report, Computer Science, Iowa State University.
3. E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, and V. Zacharias. Kaon - towards a large scale semantic web. In *K. Bauknecht, A. M. Tjoa, and G. Quirchmayr, editors, E-Commerce and Web Technologies, Third International Conference, EC-Web 2002, Aix-en-Provence, France, September 2-6, 2002, Proceedings, volume 2455 of Lecture Notes in Computer Science, pages 304-313.* Springer, 2002.
4. A. Farquhar, R. Fikes, W. Pratt, and J. Rice. Collaborative ontology construction for information integration, 1995.
5. P. Hayes, R. Saavedra, and T. Reichherzer. A collaboration development environment for ontologies. In Proceedings of the Semantic Integration Workshop, Sanibel Island, Florida, 2003.
6. N. Noy. Semantic Integration. A Survey of Ontology-based Approaches. Sigmod Record, Special Issue on Semantic Integration, 2004.
7. P. Shvaiko, J. Euzenat. Survey of Schema-based Matching Approaches. Journal on Data Semantics, 2005.
8. P. Shvaiko, F. Giunchiglia, P. Pinheiro da Silva, D. L. McGuinness. Web Explanations for Semantic Heterogeneity Discovery. In Proceedings of ESWC, 2005.
9. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. OntoEdit: Collaborative ontology development for the semantic web. In *Proceedings of the First International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Sardinia, Italia.* Springer, LNCS 2342, 2002.
10. A. Zhdanova, P. Shvaiko. Community-Driven Ontology Matching. In Proceedings of ESWC, 2006.