ISWC 2008

The 7th International Semantic Web Conference

Pavel Shvaiko
Jérôme Euzenat
Fausto Giunchiglia
Heiner Stuckenschmidt

Ontology Matching (OM 2008)

October 26, 2008

The 7th International Semantic Web Conference
October 26 – 30, 2008
Congress Center, Karlsruhe, Germany

ISWC 2008

Platinum Sponsors

Ontoprise

Gold Sponsors

BBN
eyeworkers
Microsoft
NeOn
SAP Research
Vulcan

Silver Sponsors

ACTIVE
ADUNA
Saltlux
SUPER
X-Media
Yahoo

Organizing Committee

General Chair
Tim Finin (University of Maryland, Baltimore County)

Local Chair
Rudi Studer (Universität Karlsruhe (TH), FZI Forschungszentrum Informatik)

Local Organizing Committee
Anne Eberhardt (Universität Karlsruhe)
Holger Lewen (Universität Karlsruhe)
York Sure (SAP Research Karlsruhe)

Program Chairs
Amit Sheth (Wright State University)
Steffen Staab (Universität Koblenz Landau)

Semantic Web in Use Chairs
Mike Dean (BBN)
Massimo Paolucci (DoCoMo Euro-labs)

Semantic Web Challenge Chairs
Jim Hendler (RPI, USA)
Peter Mika (Yahoo, ES)

Workshop chairs
Melliyal Annamalai (Oracle, USA)
Daniel Olmedilla (Leibniz Universität Hannover, DE)

Tutorial Chairs
Lalana Kagal (MIT)
David Martin (SRI)

Poster and Demos Chairs
Chris Bizer (Freie Universität Berlin)
Anupam Joshi (UMBC)

Doctoral Consortium Chairs
Diana Maynard (Sheffield)

Sponsor Chairs
John Domingue (The Open University)
Benjamin Grosof (Vulcan Inc.)

Metadata Chairs
Richard Cyganiak (DERI/Freie Universität Berlin)
Knud Möller (DERI)

Publicity Chair
Li Ding (RPI)

Proceedings Chair
Krishnaprasad Thirunarayan (Wright State University)

Fellowship Chair
Joel Sachs (UMBC)

Ontology Matching

OM-2008

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 an alignment, that is, a set of 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 allows the 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, rigorous and transparent evaluation of ontology matching approaches through the OAEI (Ontology Alignment Evaluation Initiative) 2008 campaign, `http://oaei.ontologymatching.org/2008`. The particular focus of this year's OAEI campaign is on real-world matching tasks from specific domains, such as cultural heritage and medicine. Moreover, there are several multi-lingual matching tasks that involve Japanese and Dutch languages. Therefore, the ontology matching evaluation initiative itself will provide a solid ground for discussion of how well the current approaches are meeting business needs.

We received 26 submissions for the technical track of the workshop. The program committee selected 6 submissions for oral presentation and 9 submissions for poster presentation. 13 matching systems participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2008.ontologymatching.org/`.

Acknowledgments. We thank all members of the program committee, authors and local organizers for their efforts. We appreciate support from the Trentino as a Lab (TasLab) project of the European Network of the Living Labs at Informatica Trentina and the OpenKnowledge European STREP (FP6-027253).

Pavel Shvaiko
Jérôme Euzenat
Fausto Giunchiglia
Heiner Stuckenschmidt

October 2008

Organization

Organizing Committee

Pavel Shvaiko, TasLab, Informatica Trentina, Trento, Italy
Jérôme Euzenat, INRIA & LIG, Grenoble, France
Fausto Giunchiglia, University of Trento, Trento, Italy
Heiner Stuckenschmidt, University of Mannheim, Mannheim, Germany

Program Committee

Olivier Bodenreider, National Library of Medicine, USA
Paolo Bouquet, University of Trento, Italy
Paolo Besana, University of Edinburgh, UK
Isabel Cruz, University of Illinois at Chicago, USA
Jérôme David, INRIA & LIG, France
Wei Hu, Southeast University, China
Ryutaro Ichise, National Institute of Informatics, Japan
Antoine Isaac, Vrije Universiteit Amsterdam, Netherlands
Anthony Jameson, DFKI, Germany
Yannis Kalfoglou, Ricoh Europe plc., UK
Vipul Kashyap, Clinical Informatics R&D, Partners HealthCare System, USA
Monika Lanzenberger, Vienna University of Technology, Austria
Patrick Lambrix, Linköpings Universitet, Sweden
Christian Meilicke, University of Mannheim, Germany
Peter Mork, The MITRE Corporation, USA
Natasha Noy, Stanford University, USA
Luigi Palopoli, University of Calabria, Italy
Ivan Pilati, TasLab, Informatica Trentina, Italy
Marco Schorlemmer, IIIA-CSIC, Spain
Luciano Serafini, FBK-IRST, Italy
Umberto Straccia, ISTI-C.N.R., Italy
Eleni Stroulia, University of Alberta, Canada
York Sure, SAP, Germany
Ludger van Elst, DFKI, Germany
Yannis Velegrakis, University of Trento, Italy
Baoshi Yan, Bosch Research, USA
Songmao Zhang, Chinese Academy of Sciences, China

Additional Reviewers

Pasquale De Meo, University Mediterranea of Reggio Calabria, Italy
Birgit Dippelreiter, Vienna University of Technology, Austria
Markus Doehring, SAP, Germany
Ji Hu, SAP, Germany
Daniel Oberle, SAP, Germany
Giovanni Quattrone, University Mediterranea of Reggio Calabria, Italy
Massimo Ruffolo, ICAR-C.N.R., Italy
Martina Seidl, Vienna University of Technology, Austria

Table of Contents

PART 1 - Technical Papers

PART 2 - OAEI Papers

PART 3 - Posters

Incoherence as a Basis for Measuring the Quality of Ontology Mappings

Christian Meilicke and Heiner Stuckenschmidt

Computer Science Institute
University of Mannheim, Germany
{christian,heiner}@informatik.uni-mannheim.de

Abstract. Traditionally, the quality of ontology matching is measured using precision and recall with respect to a reference mapping. These measures have at least two major drawbacks. First, a mapping with acceptable precision and recall might nevertheless suffer from internal logical problems that hinder a sensible use of the mapping. Second, in practical situations reference mappings are not available. To avoid these drawbacks we introduce quality measures that are based on the notion of mapping incoherence that can be used without a reference mapping. We argue that these measures are a reasonable complement to the well-known measures already used for mapping evaluation. In particular, we show that one of these measures provides a strict upper bound for the precision of a mapping.

1 Introduction

Assessing the quality of alignments is an important aspect of ontology matching. A number of different measures have been proposed for this purpose. According to [4] it can be distinguished between compliance measures, measures concerned with system usability, and performance measures that focus on runtime or memory requirements. A compliance measure compares a set of correspondences with a gold standard which should be the complete set of all correct correspondences. The most prominent compliance measures are precision and recall which have been adapted from information retrieval to the field of schema and ontology matching [1]. As complement to these measures we propose a family of measures based on the definition of mapping incoherence. These measures do not fall in one of the above mentioned categories, but should be categorized as formal or logic-based measures.

Compared to the widely used measures of precision and recall, the measures that will be proposed in this paper do not rely on the existence of a gold standard (also referred to as reference mapping). Contrary to this, they measure internal properties of a mapping based on the semantics of the ontologies aligned via the mapping. This makes our approach applicable in matching scenarios where we do not have a gold standard. Measuring the incoherence of a mapping is motivated by the idea that the incoherence of a mapping will hinder its sensible use even though it might contain a significant amount of correct correspondences. Although we introduce incoherence as a new dimension for quantifying mapping quality, there is a non-obvious relation to traditional measures. In particular, we show that we can use one of the suggested measures to compute a strict upper bound for the precision of a mapping. This result is surprising at first sight and shows the significance of the overall approach.

In the following section we discuss related work on quality measures for mappings and explain how our approach extends existing work. In section 3 we recall and refine the theory of mapping incoherence as basis for the following sections. Before introducing incoherence measures, we discuss some problems caused by mapping incoherence which justify the importance of measuring incoherence (section 4). In particular, we explain the effects of incoherence with respect to the application scenario of data transformation and query processing. In section 5 we introduce four incoherence measures divided in two groups. Measures of the first group are concerned with the impact of incoherence, while measures of the second group are used to measure the effort of repairing an incoherent mapping. Finally, we show that one of these measures can be used to compute a strict upper bound for mapping precision (section 6) followed by some concluding remarks (section 7).

2 Related Work

Several suggestions have been made to extend and introduce new evaluation measures to the field of ontology matching. In [2] Ehrig and Euzenat introduce relaxed precision and recall. Their work is motivated by the idea that a correspondence of a mapping $\mathcal{M}$ might not be totally incorrect even though it is not contained in reference mapping $\mathcal{R}$. Thus, it can be measured how close a correspondence is to a similar one in $\mathcal{R}$. Amongst others they suggest to measure the correction effort to transform such a correspondence into a correct one. We pick up this idea and suggest measuring incoherence based on the effort necessary to remove all causes of incoherence from $\mathcal{M}$. In [3] Euzenat introduces semantic precision and recall. These measures are based, roughly speaking, on comparing the (bounded) deductive closure of $\mathcal{R}$ and $\mathcal{M}$ instead of a direct comparison. Such an approach requires the use of logical reasoning where both correspondences and ontologies are considered. While Euzenat focuses on the entailment of correspondences, our approach accounts that certain combinations of correspondences result in incoherence.

Measuring mapping incoherence is closely related to measuring and repairing ontology incoherence. Thus, we adapted some of the measures defined by Qi and Hunter in [11]. Later we will show how to reduce the incoherence of a mapping $\mathcal{M}$ to concept unsatisfiability in an ontology that results from merging the ontologies matched via $\mathcal{M}$. An obvious way of measuring mapping incoherence is thus based on counting the number of unsatisfiable concepts in the merged ontology. As proposed in [6], it can be distinguished between root and derived unsatisfiable concepts. Accordant to [11], we pick up this distinction and distinguish between two types of measuring the impact of incoherence based on concept unsatisfiability.

In previous work [8, 9] we have developed and tested strategies to repair incoherent ontology mappings.[1] These strategies rely on discarding individual correspondences from an incoherent mapping $\mathcal{M}$ to finally arrive at a coherent submapping $\mathcal{M}^* \subseteq \mathcal{M}$.

[1] Notice that in previous work we misleadingly used the notion of inconsistency instead of incoherence. Precise definitions of these notions are given in [5].

Clearly, such a strategy should remove a minimal set of correspondences. This approach leads to an incoherence measure that indicates the effort of repairing incoherent mappings in numbers of correspondences to be removed. Moreover, it has been emphasized that the confidence value of a correspondence plays an important role in mapping repairing. Thus, we distinguish between a cardinality based measure and a confidence based measure. Both measures quantify the effort of revising an incoherent mapping.

3 Foundations

According to Euzenat and Shvaiko [4] a correspondence can be defined as a semantic relation between ontological entities annotated with a confidence value.

Definition 1 (Correspondence and Mapping). *Given ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, let Q be a function that defines sets of matchable elements $Q(\mathcal{O}_1)$ and $Q(\mathcal{O}_2)$. A correspondence between $\mathcal{O}_1$ and $\mathcal{O}_2$ is a 4-tuple $\langle e, e', r, n\rangle$ such that $e \in Q(\mathcal{O}_1)$ and $e' \in Q(\mathcal{O}_2)$, r is a semantic relation, and n is a confidence value from a suitable structure $\langle D, \leqslant\rangle$. A mapping between $\mathcal{O}_1$ and $\mathcal{O}_2$ is a set of correspondences between $\mathcal{O}_1$ and $\mathcal{O}_2$.*

Definition 1 allows to capture a wide class of correspondences by varying what is admissible as matchable element, semantic relation, and confidence value. In this work we consider correspondences between named concepts and properties. We also restrict correspondences to match entities of the same type, i.e. both e and e' have to be concepts or both have to be properties. We also restrict r to be $\equiv$, $\sqsubseteq$ or $\sqsupseteq$, i.e. we only focus on equivalence and subsumption correspondences. Finally, we assume that the confidence value, which can be seen as a measure of trust in the fact that the correspondence holds, is represented numerically on $D = [0.0, 1.0]$.

As argued in [8] and [9] the semantics of a mapping $\mathcal{M}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ can be defined in the context of merging $\mathcal{O}_1$ and $\mathcal{O}_2$ via $\mathcal{M}$. In this section we focus on technical aspects and postpone the discussion on adequacy and implications to the next section. A merged ontology contains the axioms of $\mathcal{O}_1$ and $\mathcal{O}_2$ as well as the correspondences of $\mathcal{M}$ translated into axioms of the merged ontology.

Definition 2 (Merged ontology). *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies (finite sets of axioms). The merged ontology $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2$ of $\mathcal{O}_1$ and $\mathcal{O}_2$ connected by $\mathcal{M}$ is defined as $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2 = \mathcal{O}1 \cup \mathcal{O}_2 \cup \{t(x) \mid x \in \mathcal{M}\}$ with t being a translation function that maps correspondences to axioms.*

Notice that in $\mathcal{O}_1$ and $\mathcal{O}_2$ a concept or property might have the same local name. To refer without ambiguity in the context of a merged ontology to an entity e which origins from $\mathcal{O}_i$ we use prefix notation $i\#e$ in the following. There is a straightforward way to translate concept correspondences and property correspondences into DL axioms. We refer to the corresponding translation function as natural translation t_n.

Definition 3 (Natural Translation). *Given correspondence $c = \langle 1\#e, 2\#e', r, n\rangle$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, the natural translation t_n of c is defined as*

$$t_n(c) \mapsto \begin{cases} 1\#e \equiv 2\#e' & \text{if } r = \equiv \\ 1\#e \sqsubseteq 2\#e' & \text{if } r = \sqsubseteq \\ 1\#e \sqsupseteq 2\#e' & \text{if } r = \sqsupseteq \end{cases}$$

Now let us briefly recall the notion of ontology incoherence. An ontology $\mathcal{O}$ is incoherent, iff there exist an unsatisfiable concept in $\mathcal{O}$. Analogous, a mapping $\mathcal{M}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$ is called incoherent due to t, if there exists an unsatisfiable concept $i\#C_{i\in\{1,2\}}$ in $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2$ that is satisfiable in $\mathcal{O}_i$. If there exists such a concept, its unsatisfiability must have (at least partially) been caused by $\mathcal{M}$.

Definition 4 (Incoherence of a Mapping). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and a translation function t. If there exists a concept $i\#C$ with $i \in \{1, 2\}$ such that $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2 \models \bot \sqsupseteq i\#C$ and $\mathcal{O}_i \not\models \bot \sqsupseteq i\#C$ then $\mathcal{M}$ is incoherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to t. Otherwise $\mathcal{M}$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to t.*

Obviously, the incoherence of a mapping is strongly affected by our choice of t. In the following we use $t = t_n$ as translation function. Notice that it is possible to define an alternative translation function. In particular, it will turn out that the measures introduced in section 5 are independent of this choice with respect to their applicability, although their results are affected by the choice of the translation function.

4 The Importance of Mapping Coherence

One might argue, that mapping coherence is only important in a very specific application scenario like reasoning in a merged ontology. In the following we show that incoherence has a negative effect on a wide range of relevant applications. In [10] four different purposes of using ontology mappings have been distinguished. A more fine-grained distinction has been proposed in [4], but most of these scenarios can be subsumed under one of these use cases.

- *Frameworks.* Mappings are described in frameworks on an abstract level independent of an intended use.
- *Terminological Reasoning.* Mappings are used to perform reasoning across aligned ontologies.
- *Data Transformation.* Data from one ontology is transferred into the terminology of another ontology based on the knowledge encoded in a mapping.
- *Query Processing.* Queries formulated with respect to a certain ontology are translated into the terminology of a different ontology.

The *Frameworks* use case is about describing mappings on an abstract level. Since we try to argue for the applicability of our approach in a practical context, it is of minor interest and will not be discussed. It is obvious that incoherence is undesirable in the *Terminological Reasoning* case as incoherence will lead to inconsistency of the whole ontology when instances are added to unsatisfiable concepts. Inconsistency, however, disables meaningful reasoning as everything can be derived from an inconsistent ontology. It is less obvious that coherence is important for the *Data Transformation* and *Query Processing* use cases. In the following, we show that an incoherent mapping will lead to serious errors in the context of data translation and query processing.

4.1 Data Transformation

To better understand the effects of incoherence in the context of *Data Transformation* let us consider the following example. Suppose there are two companies C_1 and C_2. Both use different ontologies, say $\mathcal{O}_1$ and $\mathcal{O}_2$, to describe human resources and related topics. Now it happens that C_2 takes over C_1. C_2 decides to migrate all instance data of $\mathcal{O}_1$ into $\mathcal{O}_2$. $\mathcal{O}_1$ will no longer be maintained. A terminological mapping $\mathcal{M}$ between $\mathcal{O}_1$ and $\mathcal{O}_2$ has to be created to migrate the instances of $\mathcal{O}_1$ to $\mathcal{O}_2$ in a fully automated way.

Fragments of ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ are depicted in figure 1. We refer to these fragments throughout the whole section without explicitly mentioning it in the following. *Data Transformation* can be roughly described as the following procedure.

1. For all instances a of $\mathcal{O}_1$ create a copy a' of this instance in $\mathcal{O}_2$.
2. For all concept correspondences $\langle \mathit{1\#e}, \mathit{2\#e'}, r, n\rangle \in \mathcal{M}$ with $r \in \{\equiv, \sqsubseteq\}$ and for all instances a with $\mathcal{O}_1 \models \mathit{1\#e}(a)$ add axiom $\mathit{2\#e'}(a')$ to $\mathcal{O}_2$.
3. For all property correspondences $\langle \mathit{1\#e}, \mathit{2\#e'}, r, n\rangle \in \mathcal{M}$ with $r \in \{\equiv, \sqsubseteq\}$ and for all instances a, b with $\mathcal{O}_1 \models \mathit{1\#e}(a, b)$ add axiom $\mathit{2\#e'}(a', b')$ to $\mathcal{O}_2$.

In the following we refer to the ontology resulting from migrating instances from $\mathcal{O}_i$ to $\mathcal{O}_j$ based on mapping $\mathcal{M}$ as $\mathcal{O}_j \cup \mathcal{M}(\mathcal{O}_i)$. At first sight, mapping coherence seems to be irrelevant with respect to this use case, because we do not copy any of the terminological axioms into $\mathcal{O}_j$. Consider the following correspondences to understand why this impression is deceptive.

$$\langle \mathit{1\#Person}, \mathit{2\#Person}, \equiv, 1.0\rangle \tag{1}$$

$$\langle \mathit{1\#ProjectLeader}, \mathit{2\#Project}, \sqsubseteq, 0.6\rangle \tag{2}$$

Let now mapping $\mathcal{M}$ contain correspondence (1) and (2). $\mathcal{M}$ is incoherent, due to the fact that in $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2$ concept $\mathit{1\#ProjectLeader}$ becomes unsatisfiable. Concepts $\mathit{2\#Project}$ and $\mathit{2\#Person}$ are disjoint and due to $\mathcal{M}$ concept $\mathit{1\#ProjectLeader}$ is subsumed by both of them, resulting in its unsatisfiability. Suppose now, there exists an instance a with $\mathit{1\#ProjectLeader}(a)$. Applying the migration rules results in a new instance a' with $\mathit{2\#Project}(a')$ and $\mathit{2\#Person}(a')$. Due to the disjointness of $\mathit{2\#Project}$ and $\mathit{2\#Person}$ there exists no model for $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1)$ and thus $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1)$ is an inconsistent ontology. Opposed to our first impression there seems to be a tight link between *the incoherence of* $\mathcal{M}$ and *the inconsistency of* $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1)$.

Contrary to this, mappings can be constructed, where such a direct link cannot be detected. Let $\mathcal{M}$, for example, contain correspondences (3) and (4). $\mathcal{M}$ is incoherent due to the unsatisfiability of $\mathit{2\#ProductLine}$ in the merged ontology.

$$\langle \mathit{1\#Deadline}, \mathit{2\#TimedEvent}, \sqsubseteq, 0.9\rangle \tag{3}$$

$$\langle \mathit{1\#ProjectDeadline}, \mathit{2\#ProductLine}, \sqsupseteq, 0.7\rangle \tag{4}$$

Now we have to acknowledge that both $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1)$ and $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2)$ do not become inconsistent. But what happens if we first transfer all instances x of $\mathcal{O}_2$ to $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2)$ and then again transfer the x' instances of $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2)$ to $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2))$?

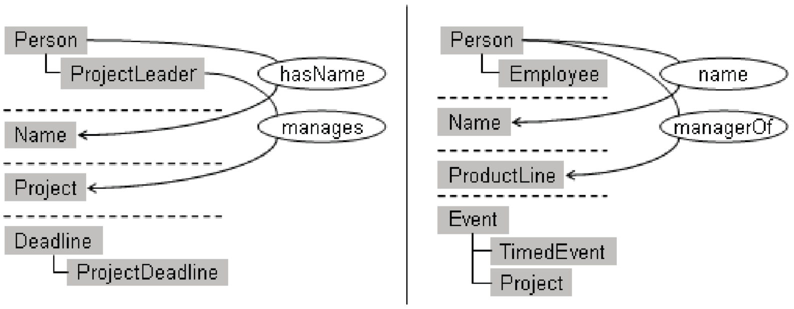


Fig. 1. Fragments of ontologies $\mathcal{O}_1$ (on the left) and $\mathcal{O}_2$ (on the right). A square represents a concept, an ellipse a property, subsumption is represented by indentation. Domain and range of a property are restricted to be the concepts connected by the accordant arrow. Dashed horizontal lines represent disjointness between concepts.

Given some instance a with $\mathcal{O}_2 \models \mathit{2\#ProductLine}(a)$ after the first step we have the counterpart of a, namely a', with $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2) \models \mathit{1\#ProjectDeadline}(a')$ by applying correspondence (4) and can derive $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2) \models \mathit{1\#Deadline}(a')$. After the second step we have $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2)) \models \mathit{2\#TimedEvent}(a'')$ by applying correspondence (3). We expect that adding axiom $a = a''$ does not affect the consistency of $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2))$. But $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2))$ now implies $\mathit{2\#ProductLine}(a)$ as well as $\mathit{2\#Event}(a)$. Since $\mathit{2\#ProductLine}$ and $\mathit{2\#Event}$ are defined to be disjoint, there exists no model for $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2))$. Again, we find a strong link between mapping incoherence and inconsistency after instance migration.

4.2 Query Processing

In the following we revisit a variant of the example given above to better understand the use case of *Query Processing*. Again, company C_2 takes over C_1. But this time both $\mathcal{O}_1$ and $\mathcal{O}_2$ are maintained. Instead of migrating all instances from $\mathcal{O}_1$ to $\mathcal{O}_2$ queries are rewritten at runtime to enable information integration between $\mathcal{O}_1$ and $\mathcal{O}_2$. A terminological mapping is the key for information integration. It is used for processing queries and generating result sets which contain data from both ontologies. As we are concerned with theoretical issues, we argue on an abstract level instead of discussing e.g. characteristics of a SPARQL implementation. A query language for DL based knowledge bases should at least support instance retrieval for complex concept descriptions. Depending on the concrete query language there might be a complex set of rewriting rules. At least, it must contain variants of the following two rules.

R_1: Let $i\#C$ and $i\#D$ be concept descriptions in the language of $\mathcal{O}_i$. If $\mathcal{O}_i \models i\#C \equiv i\#D$, then query q can be transformed into an equivalent query by replacing all occurrences of $i\#C$ by $i\#D$.

R_2: Let $i\#C$ and $j\#D$ be concept descriptions in the language of $\mathcal{O}_i$, respectively $\mathcal{O}_j$. If there exists a correspondence $\langle i\#C, j\#D, \equiv, n\rangle \in \mathcal{M}$, then query q can be transformed into an equivalent query by replacing all occurrences of $i\#C$ by $j\#D$.

Suppose we query for the name of all project leaders, formally speaking we are interested in the instances of $\exists \mathit{1\#hasName}^{-1}.\mathit{1\#ProjectLeader}$. To receive instances of both $\mathcal{O}_1$ and $\mathcal{O}_2$ we have to rewrite the query for $\mathcal{O}_2$. Now let $\mathcal{M}$ contain correspondences (5), (6), and (7).

$$\langle \mathit{1\#hasName}, \mathit{2\#name}, \equiv, 0.9 \rangle \tag{5}$$

$$\langle \mathit{1\#Project}, \mathit{2\#Project}, \equiv, 1.0 \rangle \tag{6}$$

$$\langle \mathit{1\#manages}, \mathit{2\#managerOf}, \equiv, 0.7 \rangle \tag{7}$$

Suppose that $\mathcal{O}_1$ contains axiom $\mathit{1\#ProjectLeader} \equiv \exists \mathit{1\#manages}.\mathit{1\#Project}$. We exploit this axiom by applying R_1. Now for every concept and property name that occurs in $\exists \mathit{1\#hasName}^{-1}.\exists \mathit{1\#manages}.\mathit{1\#Project}$, there exists a direct counterpart in $\mathcal{O}_2$ specified in $\mathcal{M}$. By applying R_2 we thus finally end with a concept description in the language of $\mathcal{O}_2$.

$$\exists \mathit{1\#hasName}^{-1}.\mathit{1\#ProjectLeader} \tag{8}$$

$$\overset{R_1}{\Longleftrightarrow} \exists \mathit{1\#hasName}^{-1}.\exists \mathit{1\#manages}.\mathit{1\#Project} \tag{9}$$

$$\overset{R_2}{\Longleftrightarrow} \exists \mathit{2\#name}^{-1}.\exists \mathit{2\#managerOf}.\mathit{2\#Project} \tag{10}$$

What happens if we process the query based on this concept description to $\mathcal{O}_2$? As result we receive the empty set. The range of $\mathit{2\#managerOf}$ is concept $\mathit{2\#ProductLine}$, and $\mathit{2\#ProductLine}$ is defined to be disjoint with $\mathit{2\#Project}$. Thus, for logical reasons there exists no instance of concept description (10) in $\mathcal{O}_2$.

This problem is obviously caused by the incorrectness of correspondence (7). But the incorrectness of (7) does not only affect the query under discussion. It also causes mapping $\mathcal{M}$ to become incoherent, because in the merged ontology $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2$ concept $\mathit{1\#ProjectLeader}$ becomes unsatisfiable due to its equivalence with concept description $\exists \mathit{1\#manages}.\mathit{1\#Project}$. This time we find a strong link between *mapping incoherency* and the *incorrectness of a query result* due to processing the mapping.

5 Measuring Incoherence

The definition of mapping incoherence given above is a boolean criterion that only distinguishes between coherent and incoherent mappings. Contrary to this, an incoherence measure should satisfy $m(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) > m(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}')$ if $\mathcal{M}$ has *a higher degree of incoherence* than $\mathcal{M}'$. At the moment we might have an intuitive understanding of different degrees of incoherence, but a precise definition has to be given in the following subsections. Up to now, we define an incoherence measure to satisfy the following constraint.

Definition 5 (Incoherence Measure). *Let $\mathcal{M}$ be a mapping between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and let t be an translation function. An incoherence measure m^t maps $\mathcal{O}_1, \mathcal{O}_2$, and $\mathcal{M}$ to a value in $[0, 1]$ such that $m^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = 0$ iff $\mathcal{M}$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to t.*

In the following we distinguish between effect-based and revision-based measures. The former are concerned with the negative impact of mapping incoherence. The latter measure the effort necessary to revise a mapping by removing incoherences. Both approaches make it possible to extend the boolean property of incoherence to a continuous measure of its degree.

5.1 Measuring the Impact of Incoherence

The first measure to be introduced is based on the idea of counting unsatisfiable concepts. It is an adaption of an ontology incoherence measure introduced in [11]. Before we proceed, we need to agree on some abbreviations and naming conventions.

Definition 6. *Let $\mathcal{O}$ be an ontology. Then $\mathit{CO}(\mathcal{O})$ refers to the set of named concepts in $\mathcal{O}$ and $\mathit{US}(\mathcal{O}) = \{C \in \mathit{CO}(\mathcal{O}) \mid \mathcal{O} \models C \sqsubseteq \bot\}$ refers to the set of unsatisfiable concepts in $\mathcal{O}$.*

Contrary to measuring incoherences in ontologies, we have to distinguish between two types of concept unsatisfiability in the merged ontology: There are unsatisfiable concepts in $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2$ which have already been unsatisfiable in $\mathcal{O}_1$, respectively $\mathcal{O}_2$, while there are unsatisfiable concepts which have been satisfiable in $\mathcal{O}_1$, respectively $\mathcal{O}_2$. We are interested in the latter concepts. In particular, we compare the number of these concepts with the number of all named concepts satisfiable in $\mathcal{O}_1$ or $\mathcal{O}_2$.

Definition 7 (Unsatisfiability Measure). *Let $\mathcal{M}$ be a mapping between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, and let t be a translation function. Unsatisfiability measure m^t_{sat} is defined by*

$$m^t_{sat}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = \frac{|\mathit{US}(\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2) \setminus (\mathit{US}(\mathcal{O}_1) \cup \mathit{US}(\mathcal{O}_2))|}{|\mathit{CO}(\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2) \setminus (\mathit{US}(\mathcal{O}_1) \cup \mathit{US}(\mathcal{O}_2))|}$$

This measure can be criticised for the following reason. Suppose again, we have an incoherent mapping $\mathcal{M}$ for ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ depicted in figure 1. Suppose that due to $\mathcal{M}$ concept *1#Person* becomes unsatisfiable in the merged ontology. As a consequence concept *1#ProjectLeader* becomes unsatisfiable, too. By applying definition 7 we thus measure both direct impact (unsatisfiability of *1#Person*) and indirect impact (unsatisfiability of *1#ProjectLeader*) of $\mathcal{M}$, even though we might only be interested in the direct impact. The distinction between root and derived unsatisfiability, as introduced in [6], solves this problem. A precise definition requires us to recall the notion of a MUPS, defined in [12], which is minimal unsatisfiability preserving sub-TBox.

Definition 8 (MUPS). *Let $\mathcal{O}$ be an ontology, let $\mathcal{T} \subseteq \mathcal{O}$ be the TBox of $\mathcal{O}$, and let $C \in \mathit{US}(\mathcal{O})$. A set $\mathcal{T}' \subseteq \mathcal{T}$ is a minimal unsatisfiability preserving sub-TBox (MUPS) in $\mathcal{T}$ for C if C is unsatisfiable in $\mathcal{T}'$ and C is satisfiable in every $\mathcal{T}'' \subset \mathcal{T}'$. The set of all MUPS with respect to C is referred to as* mups$(\mathcal{O}, C)$.

A MUPS for a concept C can be seen as a minimal explanation of its unsatisfiability. Whenever there exists another unsatisfiable concept D such that the minimal explanation of C's unsatisfiability also explains the unsatisfiability of D then C is referred to as derived unsatisfiable concept, because one reason for C's unsatisfiability is the unsatisfiability of D.

Definition 9 (Derived and Root Unsatisfiability). *Let $\mathcal{O}$ be an ontology and let $C \in US(\mathcal{O})$. C is a derived unsatisfiable concept if there exists $D \neq C \in US(\mathcal{O})$ such that there exist $M \in$ mups$(\mathcal{O}, C)$ and $M' \in$ mups$(\mathcal{O}, D)$ with $M \supseteq M'$. Otherwise C is a root unsatisfiable concept. The set of derived unsatisfiable concepts of $\mathcal{O}$ is referred to as $US_D(\mathcal{O})$ and the set of root unsatisfiable concepts is referred to as $US_R(\mathcal{O})$.*

Similar to the unsatisfiability measure we can now introduce the root unsatisfiability measure by considering only root unsatisfiable concepts instead of all unsatisfiable concepts in the merged ontology.

Definition 10 (Root Unsatisfiability Measure). *Let $\mathcal{M}$ be a mapping between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, and let t be a translation function. Root unsatisfiability measure m^t_{rsat} is defined by*

$$m^t_{rsat}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = \frac{|US_R(\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2) \setminus (US(\mathcal{O}_1) \cup US(\mathcal{O}_2))|}{|CO(\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2) \setminus (US(\mathcal{O}_1) \cup US(\mathcal{O}_2))|}$$

Obviously, we have $m^t_{sat}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) \geq m^t_{rsat}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M})$ for each mapping $\mathcal{M}$ between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. As argued above, the m^t_{rsat} measure has to be preferred. Nevertheless, a non trivial algorithm is required to compute the set of root unsatisfiable concepts as described in [6] which makes the application of this measure more expensive from a computational point of view.

5.2 Measuring the Effort of Mapping Revision

The second type of measure is concerned with the effort of revising incoherent mappings. We use the term revision to describe the process of removing correspondences from an incoherent mapping until a coherent submapping has been found. If a revision is conducted by a domain expert we would, for example, be able to measure the time necessary. Since we want our measure to be computable without any human intervention, we thus have to think of an automated strategy to revise a mapping. Such a strategy should obviously remove a minimum number of correspondences, because we would like to keep as much information in the mapping as possible. The following measure is based on this idea and compares the number of correspondences that would be removed by such a strategy with the number of all correspondences in the mapping.

Definition 11 (Maximum Cardinality Measure). *Let $\mathcal{M}$ be a mapping between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, and let t be a translation function. Maximum cardinality measure m^t_{card} is defined by*

$$m^t_{card}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = \frac{|\mathcal{M} \setminus \mathcal{M}'|}{|\mathcal{M}|}$$

where $\mathcal{M}' \subseteq \mathcal{M}$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to t and there exists no $\mathcal{M}'' \subseteq \mathcal{M}$ with $|\mathcal{M}''| > |\mathcal{M}'|$ such that $\mathcal{M}''$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to t.

Suppose there are incoherent mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ with $|\mathcal{M}_1| = |\mathcal{M}_2| = 10$. Further suppose, according to the naming convention in definition 11, we have $|\mathcal{M}'_1| = 8$

and $|\mathcal{M}'_2| = 7$. Thus, we have $m^t_{card}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}_1) = 0.2$ and $m^t_{card}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}_2) = 0.3$. But now suppose that all of the correspondences in $\mathcal{M}_1$ have the same confidence value, say 1. Contrary to this, $\mathcal{M}_2$ differs with respect to the confidence value of its correspondences. In particular, it turns out that all of the three correspondences that have been removed have a very low confidence value compared to the remaining correspondences. The following definition introduces the maximum trust measure which is similar to the maximum cardinality measure but accounts for differences in the confidence distribution.

Definition 12 (Maximum Trust Measure). *Let $\mathcal{M}$ be a mapping between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, and let t be a translation function. Further, let $conf : \mathcal{M} \rightarrow [0, 1]$ be a function that maps a correspondence on its confidence value. Maximum trust measure m^t_{trust} is defined by*

$$m^t_{trust}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = \frac{\sum\limits_{c \in \mathcal{M} \setminus \mathcal{M}'} conf(c)}{\sum\limits_{c \in \mathcal{M}} conf(c)}$$

where $\mathcal{M}' \subseteq \mathcal{M}$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to t and there exists no $\mathcal{M}'' \subseteq \mathcal{M}$ with $\sum_{c \in \mathcal{M}''} conf(c) > \sum_{c \in \mathcal{M}'} conf(c)$ such that $\mathcal{M}''$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to t.

This measure is derived from the algorithm already described in [9] which can be used to compute $\mathcal{M}'$ for a specific type of mappings. Namely, one-to-one mappings that contain only correspondences expressing equivalences between concepts. We also used it in the context of mapping extraction [8]. Its application is motivated by the idea that $\mathcal{M} \setminus \mathcal{M}'$ mainly contains incorrect correspondences given an appropriate confidence distribution. We adapted this idea to introduce the m^t_{trust} measure as confidence weighted complement to the m^t_{card} measure.

Notice that computing both of these measures requires to solve computational hard problems. On the one hand only full-fledged reasoning guarantees completeness in detecting unsatisfiability. On the other hand the underlying problem is the optimization problem of finding a hitting set $H \subseteq \mathcal{M}$ of minimal cardinality (respectively minimal confidence total) over the set all of minimal incoherent subsets of $\mathcal{M}$. This problem is known to be NP-complete [7]. Nevertheless, first experiments indicate that both measures can be computed in acceptable time for small and medium sized ontologies.

6 Truth and Coherence

In the following we are concerned with an important interrelation between the classical compliance measure of precision and the maximum cardinality measure of incoherence. Philosophically speaking, we are interested in how far an incoherent mapping can truly express semantic relations between ontological entities. Accordant to [1], the precision of a mapping can be defined as follows.

Definition 13 (Precision). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, let $\mathcal{R}$ be a reference mapping between $\mathcal{O}_1$ and $\mathcal{O}_2$. The precision of $\mathcal{M}$ with respect to $\mathcal{R}$ is defined as $precision(\mathcal{M}, \mathcal{R}) = |\mathcal{M} \cap \mathcal{R}| \,/\, |\mathcal{M}|$.*

In section 4 we argued that an incoherent mapping $\mathcal{M}$ causes different kinds of problems when applied in a realistic scenario. More precisely, it is the incorrectness of a correspondence that causes both the problem as well as the incoherence. Even though incorrect correspondences not necessarily result in incoherence, we can be sure that an incoherent mapping contains at least one incorrect correspondence. Thus, a well-modeled reference mapping $\mathcal{R}$ will be coherent due to each translation function t compatible with the mapping semantics accepted by the person who created $\mathcal{R}$. The following proposition corresponds to this consideration.

Proposition 1 (Incoherence and Precision). *Let $\mathcal{R}$ be a reference mapping between $\mathcal{O}_1$ and $\mathcal{O}_2$. Further let mapping $\mathcal{M}$ be incoherent and let $\mathcal{R}$ be coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to translation function t. Then we have $precision(\mathcal{M}, \mathcal{R}) < 1$.*

Proof. Given the coherence of $\mathcal{R}$, it can be concluded that every subset of $\mathcal{R}$ is coherent, too. Since $\mathcal{M}$ is incoherent, it is thus no subset of $\mathcal{R}$, i.e $\mathcal{M} \setminus \mathcal{R} \neq \emptyset$. We conclude that $\mathcal{M} \cap \mathcal{R} \subset \mathcal{M}$. It follows directly $precision(\mathcal{M}, \mathcal{R}) < 1$.

Notice that automatically generated mappings normally do not have a precision of 1. Thus, the application of proposition 1 is only of limited benefit. Nevertheless, it can be generalized in a non trivial way by exploiting the definition of the maximum cardinality measure (definition 11). This generalization allows us to compute a non trivial upper bound for mapping precision without any knowledge of $\mathcal{R}$.

Proposition 2 (Upper Bound for Precision). *Let $\mathcal{M}$ be a mapping and $\mathcal{R}$ be a reference mapping between $\mathcal{O}_1$ and $\mathcal{O}_2$. Further let $\mathcal{R}$ be coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to translation function t. Then we have $precision(\mathcal{M}, \mathcal{R}) \leq 1 - m^t_{card}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M})$.*

Proof. Accordant to definition 11 let $\mathcal{M}' \subseteq \mathcal{M}$ be the coherent subset of $\mathcal{M}$ with maximum cardinality. Further let be $\mathcal{M}^* = \mathcal{M} \cap \mathcal{R}$, i.e. $\mathcal{M}^*$ consist of all correct correspondences in $\mathcal{M}$. Since $\mathcal{M}^*$ is a subset of $\mathcal{R}$ and $\mathcal{R}$ is coherent with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$ due to t, we conclude that $\mathcal{M}^*$ is also coherent. It follows that $|\mathcal{M}^*| \leq |\mathcal{M}'|$, because otherwise $\mathcal{M}'$ would not be the coherent submapping of maximum cardinality contrary to definition 11. In summary, the following inequation holds.

$$precision(\mathcal{M}, \mathcal{R}) = \frac{|\mathcal{M} \cap \mathcal{R}|}{|\mathcal{M}|} = \frac{|\mathcal{M}^*|}{|\mathcal{M}|} < \frac{|\mathcal{M}'|}{|\mathcal{M}|} = 1 - \frac{|\mathcal{M}' \setminus \mathcal{M}|}{|\mathcal{M}|} = m^t_{card}(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M})$$

Proposition 2 reveals an important interrelation between coherence and precision. The counterpart of mapping precision is the measure of recall. At first glimpse it seems that recall and coherence describe independent properties of a mapping. Nevertheless, there exists a non trivial relation between recall and coherence that allows to derive comparative statements about the relative recall of two overlapping mappings in some cases. Although this interrelation is not as significant as the the one expressed in proposition 2, further theoretical considerations are required.

The utility of proposition 2 in the evaluation process essentially depends on the distance between the upper bound and the actual value of mapping precision. In particular, measuring low values for the m^t_{card} measure will lead to a poor differentiation with respect to the precision that has to be expected. In initial experiments, not included in this paper due to lack of space, first results indicate that the upper bound for mapping precision strongly varies and can be used to filter out highly imprecise mappings.

7 Conclusion

In this paper, we have discussed the notion of incoherence of ontology mappings and its role in the assessment of automatically created mappings. There are two main conclusions of this work. First, we conclude that incoherence is an important aspect of mapping quality as incoherent mappings have undesirable effects on most relevant application scenarios as we have demonstrated in section 4. Second, appropriate measures of incoherence can help to assess the quality of a mapping even if no reference mapping is available and thus precision and recall cannot be determined. In particular, the measure of incoherence provided in definition 11 provides a strict upper bound for the precision of a mapping and can therefore be used as a guideline for estimating the performance of matching systems. In future work experimental studies will show in how far the proposed measures can be effectively applied in the evaluation process.[2]

Acknowledgement The work has been partially supported by the German Science Foundation (DFG) in the Emmy Noether Programme under contract STU 266/3-1.

References

1. Hong-Hai Do, Sergey Melnik, and Erhard Rahm. Comparison of schema matching evaluations. In *Proc. of the GI-Workshop Web and Databases*, Erfurt, Germany, 2002.
2. Marc Ehrig and Jerome Euzenat. Relaxed precision and recall for ontology matching. In *Proc. of the K-Cap 2005 Workshop on Integrating Ontology*, Banff, Canada, 2005.
3. Jerome Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proc. of th 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007.
4. Jerome Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.
5. Peter Haase and Guilin Qi. An analysis of approaches to resolving inconsistencies in DL-based ontologies. In *Proc. of the International Workshop on Ontology Dynamics*, Innsbruck, Austria, 2007.
6. Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and James Hendler. Debugging unsatisfiable classes in OWL ontologies. *Journal of Web Semantics*, 2005.
7. Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Plenum, 1972.
8. Christian Meilicke and Heiner Stuckenschmidt. Analyzing mapping extraction approaches. In *Proc. of the ISWC 2007 Workshop on Ontology Matching*, Busan, Korea, 2007.
9. Christian Meilicke and Heiner Stuckenschmidt. Applying logical constraints to ontology matching. In *Proc. of the 30th German Conference on Artificial Intelligence*, Osnabrück, Germany, 2007.
10. Natasha Noy and Heiner Stuckenschmidt. Ontology alignment: An annotated bibliography. In *Semantic Interoperability and Integration*, Dagstuhl, Germany, 2005.
11. Guilin Qi and Anthony Hunter. Measuring incoherence in description logic-based ontologies. In *Proc. of the 6th International Semantic Web Conference*, 2007.
12. Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. of 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

[2] A first beta-version of our system supporting the measures m^t_{sat}, m^t_{card}, and m^t_{trust} based on a slightly modified natural translation can be obtained on request.

Resolution of conflicts among ontology mappings: a fuzzy approach $^{(*)}$

Alfio Ferrara[1], Davide Lorusso[1],
Giorgos Stamou[2], Giorgos Stoilos[2], Vassilis Tzouvaras[2], Tassos Venetis[2]

[1] Università degli Studi di Milano,
DICo, via Comelico 39, 20135 Milano, Italy,
{ferrara, lorusso}@dico.unimi.it

[2] National Technical University of Athens,
IVML, Iroon Polytechniou 9, 15780 Athens, Greece,
{gstam,gstoil,tzouvaras,avenet}@image.ece.ntua.gr

Abstract. Interoperability is a strong requirement in open distributed systems and in the Semantic Web. The need for ontology integration is not always completely met by the available ontology matching techniques because, in most cases, the semantics of the compared ontologies is not considered, thus leading to inconsistent mappings. Probabilistic approaches has been proposed to validate mappings and solve the inconsistencies, based on a mapping confidence measure. As probabilistic approaches suffer from the lack of well-founded likelihood measures of mapping correctness, we propose a validation approach based on fuzzy interpretation of mappings, which better models the notion of degree of similarity between ontology elements. Moreover, we describe a conflict resolution method which computes the minimal sets of conflicting mappings and can be the ground of different validation strategies.

1 Introduction

In the context of the Semantic Web, the available information is organized in ontologies. Ontologies are controlled vocabularies describing objects and relations between them in a formal way, and have a grammar for using the vocabulary terms in order to express something meaningful within a specified domain of interest. However, ontologies themselves can be heterogeneous: given two ontologies describing a reference domain, the same real entity can be denoted in the two ontologies with different names or it can be defined in different ways (an entity of one ontology may be the union of two of the entities of the other ontology) whereas both ontologies may be expressed in different languages, though expressing the same knowledge. In order to achieve the goal of ontology interoperability, we need to align heterogeneous ontologies by (semi-)automatically discovering mappings between the elements in two different ontologies. Most of

$^{(*)}$ This paper has been partially funded by the BOEMIE Project, FP6-027538, 6th EU Framework Programme.

the existing matching techniques do not take into account the semantics of the compared ontologies, therefore the resulting mappings can not be interpreted as semantic relations among the ontology elements, which is a necessary condition to perform integration and, subsequently, query answering over the integrated schema. Recently, several studies have focused on mapping validation with respect to the semantics of the ontologies involved and, at the same time, by maintaining the uncertain nature of mappings. In [1] is proposed a language for representation and reasoning with uncertain mappings by combining ontology and rule languages with probabilistic reasoning. This method represents confidence values as error probabilities in order to resolve inconsistencies by using trust probabilities, and to reason about these on a numeric level. In our previous work [2] we presented a tool for mapping validation with the help of probabilistic reasoning. The idea is to assume a semantic interpretation of ontology mappings as probabilistic and hypothetical relations among ontology elements in order to build a unique distributed knowledge base from the two independent ontologies and, subsequently, check for inconsistencies.

Probabilistic approaches for mapping validation suffer of limitations due to the nature of mappings and the way the probability values are computed. Our idea is to adopt a completely different interpretation in order to be able to validate mappings even in the absence of a precise semantics and in the presence of uncertainty. Assuming that an ontology mapping states the generic similarity of two concepts, we can assert that the objects modeled by the first concept can be also modeled by the second concept to a certain degree. In other words, the individuals of the first concept belong to the second concept with a certain degree, which is exactly the semantics of fuzzy membership functions. The degree of membership is determined by the strength of the similarity relation, computed by the same matching technique which produced the mapping. By using the acquired mappings to create fuzzy individual assertions, we provide a formal interpretation of mappings. Moreover, on the grounds of the Fuzzy Description Logics theory, we are able to perform reasoning on the integrated ontologies in order to detect and solve inconsistencies by mapping refinement, which is another difference compared to [2].

2 Ontology Mappings and Fuzzy Interpretation

In this section, we provide an introduction to a fuzzy extension of Description Logics (DL) by adding degrees to DL facts; we call this extension f-DL. This extension is based on Fuzzy Sets and Fuzzy Logic [3] and on previous work on fuzzy Description Logics [4, 5].

As usual fuzzy DLs are defined by an alphabet of distinct *concept names* (*class names*) $\mathbf{C}$, *role names* (*property names*) $\mathbf{R}$ and *individuals* $\mathbf{I}$. The set of roles (properties) is defined as $\mathbf{R} \cup \{R^- \mid R \in \mathbf{R}\}$, where R^- represents the *inverse* of R. Elementary descriptions are *atomic concepts* and *atomic roles*, and by using concept constructors we can define complex concept descriptions. More precisely, if $A, C, D \in \mathbf{C}$, $R, S \in \mathbf{R}$ and $p \in \mathbb{N}$, where A is an atomic concept,

C, D are complex concepts, and S is an atomic role [6], then f-$\mathcal{SHIN}$-concepts are defined inductively by the following abstract syntax:

$$C, D \longrightarrow \bot \mid \top \mid A \mid C \sqcup D \mid C \sqcap D \mid \neg C \mid \forall R.C \mid \exists R.C \mid \geq pS \mid \leq pS$$

A fuzzy DL Knowledge Base Σ is a triple $\Sigma = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a *TBox*, $\mathcal{R}$ a *RBox* and $\mathcal{A}$ an *ABox*. A TBox is a set of *concept subsumption axioms* of the form, $C \sqsubseteq D$ and *concept equivalence axioms* of the form $C \equiv D$, where C, D are f-$\mathcal{SHIN}$-concepts. An RBox is a set of *transitive role axioms* of the form $\mathsf{Trans}(R)$ and *role subsumption axioms* of the form $R \sqsubseteq S$, where R, S are f-$\mathcal{SHIN}$-roles, while an ABox is a set of *fuzzy concept* and *fuzzy role assertions* of the form $(a : C) \bowtie n$ and $((a, b) : R) \bowtie n$, or individual equalities and inequalities of the form $a = b$ or $a \neq b$, where $a, b \in \mathbf{I}$, $\bowtie \in \{\geq, >, \leq, <\}$ and $n \in [0, 1]$.

The semantics of f-DL are based on *fuzzy interpretations*. A fuzzy interpretation $\mathcal{I}$ is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the domain $\Delta^{\mathcal{I}}$ is, like the crisp case, a non-empty set of objects and $\cdot^{\mathcal{I}}$ is a fuzzy interpretation function, which maps

- an *individual name* o to an object $o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$,
- a *concept name* C to a membership function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow [0, 1]$ [1], and
- a *property name* R to a membership function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$.

Complex f-$\mathcal{SHIN}$-concepts, roles and axioms are interpreted by extending fuzzy interpretation, making use of fuzzy set theoretic operators and notions, like subsethood, from the fuzzy set literature. The complete semantics are presented in Table 1, where sup is the *supremum*, inf is the *infimum*, c is a *fuzzy complement*, t is a *fuzzy conjunction* (*t-norm*), u is a *fuzzy disjunction* (*t-conorm*) and $\mathcal{J}$ is a *fuzzy implication*.

A fuzzy knowledge base Σ is satisfiable iff there exists a fuzzy interpretation $\mathcal{I}$ which satisfies all axioms in Σ. Basic inference problems in f-DL are: (i) check if a fuzzy knowledge base is *consistent* i.e. has a model, (ii) check if D *subsumes* C w.r.t. Σ, i.e. $\Sigma \models C \sqsubseteq D$, (iii) check if a is an instance of C to degree $\bowtie n$, i.e. $\Sigma \models a : C \bowtie n$, where $\bowtie \in \{\geq, >, \leq, <\}$ and (iv) determine the *greatest lower bound* of a w.r.t. Σ, denoted $glb(\Sigma, a)$, where $glb(\Sigma, a) = \sup\{n \mid \Sigma \models a \geq n\}$.

2.1 Fuzzy Interpretation of Ontology Mappings

In order to achieve ontology interoperability heterogeneous ontologies should be (semi-)automatically aligned. The problem called "Ontology Alignment" or "Ontology Matching" can be described as follows: given two ontologies each describing a set of discrete entities (which can be classes, properties, predicates, etc.), find the relationships (e.g., equivalence or subsumption) that hold between these entities. In a more formal way we could say that a mapping $\mathcal{M}$ is a set of tuples

$$m_i = \langle C_i, C'_i, n_i, R_i \rangle$$

for $i \in I$, where

[1] For instance, given an object $a \in \Delta^{\mathcal{I}}$ and a class name C, $C^{\mathcal{I}}(a)$ gives a degree of confidence (such as 0.8) that the object a belongs to the fuzzy concept C.

Table 1. Fuzzy DL Descriptions and Axioms

Abstract Syntax	DL Syntax	Semantics
Bottom	$\bot$	$\bot^{\mathcal{I}}(a) = 0$
Top	$\top$	$\top^{\mathcal{I}}(a) = 1$
Intersection	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}}(a) = t(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
Union	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}}(a) = u(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
Complement	$\neg C$	$(\neg C)^{I}(a) = c(C^{\mathcal{I}}(a))$
Existential Restriction	$\exists R.C$	$(\exists R.C)^{\mathcal{I}}(a) = \sup_{b \in \Delta^{\mathcal{I}}} t(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))$
Universal Restriction	$\forall R.C$	$(\forall R.C)^{\mathcal{I}}(a) = \inf_{b \in \Delta^{\mathcal{I}}} J(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))$
Min Cardinality Restriction	$\geq nR$	$(\geq nR)^{\mathcal{I}}(a) = \sup\limits_{b_1,\ldots,b_p \in \Delta^{\mathcal{I}}} t(\mathop{\mathrm{t}}\limits_{i=1}^{p} R^{\mathcal{I}}(a,b_i), \mathop{\mathrm{t}}\limits_{i<j} \{b_i \neq b_j\})$
Max Cardinality Restriction	$\leq nR$	$(\leq nR)^{\mathcal{I}}(a) = \inf\limits_{b_1,\ldots,b_{p+1} \in \Delta^{\mathcal{I}}} \mathcal{J}(\mathop{\mathrm{t}}\limits_{i=1}^{p+1} R^{\mathcal{I}}(a,b_i), \mathop{\mathrm{u}}\limits_{i<j} \{b_i = b_j\})$
SubClass	$C \sqsubseteq D$	$C^{\mathcal{I}}(a) \leq D^{\mathcal{I}}(a)$
Equivalent Classes	$C \equiv D$	$C^{\mathcal{I}}(a) = D\mathcal{I}(a)$
SubRole	$R \sqsubseteq S$	$R^{\mathcal{I}}(a,b) \leq S^{\mathcal{I}}(a,b)$
Class Individual	$o : C \bowtie n$	$C^{\mathcal{I}}(o^{\mathcal{I}}) \bowtie n$
Role Individual	$(o, o') : R \bowtie n$	$R^{\mathcal{I}}(o^{\mathcal{I}}, o'^{\mathcal{I}}) \bowtie n$
Disjoint Classes	$C \sqsubseteq \neg D$	$C^{\mathcal{I}}(a) \leq 1 - D^{\mathcal{I}}(a)$
Transitive Object Property	$\mathsf{Trans}(R)$	$\sup_{b \in \Delta^{\mathcal{I}}} t(R^{\mathcal{I}}(a,b), R^{\mathcal{I}}(b,c)) \leq R^{\mathcal{I}}(a,c)$

- C_i, C'_i are the discrete entities from two ontologies, $\mathcal{O}$ and $\mathcal{O}$', between which a relation is asserted by the mapping;
- n_i is a value, which is a part of structure $\langle \mathcal{D}, \leq, 0, 1 \rangle$, where $\mathcal{D}$ is the set of degrees and $\forall d \in \mathcal{D}, 0 \leq d \leq 1$ holds, that denotes the strength of the relation R_i;
- and R_i is one of the following relations $R = \{\equiv, \sqsubseteq, \sqsupseteq\}$, that holds between the entities C_i and C'_i.

Another way to represent these relations using bridge rules, as used in distributed description logics [7], is

$$C_i \xrightarrow{\equiv} C'_i : n \qquad C_i \xrightarrow{\sqsubseteq} C'_i : n \qquad C_i \xrightarrow{\sqsupseteq} C'_i : n$$

In order to take into account the uncertain and fuzzy nature of the mappings we define a fuzzy mapping as follows.

Definition 1 (Fuzzy Mapping). *Given two ontology elements C_i and C'_i, a fuzzy mapping $fm_i = \langle C_i, C'_i, n_i, R_i \rangle$ is a mapping m_i, whose value n_i denotes the degree that the semantic relation R_i holds between C_i and C'_i, where R_i can be one of equivalence ($\equiv$) or subsumption ($\sqsubseteq$, $\sqsupseteq$).*

This way the mappings are formalized as fuzzy knowledge. The basic idea behind the formalization of mappings as fuzzy knowledge is to use the mappings so as to create fuzzy individual assertions. In order to do that we must provide semantics for the mappings and to do so we will use the Fuzzy Set Theory [3]. Let $\mathcal{I}$ be a fuzzy interpretation, while let $\mathcal{I}_c$ be a crisp interpretation. Then we have the following conditions:

$$\mathcal{I} \models C_i \xrightarrow{\equiv} C_i' : n_i \Longleftrightarrow \forall b.b \in C_i^{\mathcal{I}_c} \rightarrow C_i'^{\mathcal{I}}(b) = n_i$$
$$\mathcal{I} \models C_i \xrightarrow{\sqsubseteq} C_i' : n_i \Longleftrightarrow \forall b.b \in C_i^{\mathcal{I}_c} \rightarrow C_i'^{\mathcal{I}}(b) \geq n_i$$
$$\mathcal{I} \models C_i \xrightarrow{\sqsupseteq} C_i' : n_i \Longleftrightarrow \forall b.b \in C_i^{\mathcal{I}_c} \rightarrow C_i'^{\mathcal{I}}(b) \leq n_i$$

The above definitions imply a procedure by which we can transfer individuals from the source ontology $\mathcal{O}$ to the target ontology $\mathcal{O}$', creating a set of fuzzy assertions A_M. This procedure will be described in more detail in the following.

2.2 Fuzzy DL Reasoning with FiRE

In this section we provide a short introduction to the Fuzzy Reasoning Engine FiRE [8]. FiRE is a prototype JAVA implementation of a fuzzy algorithm for an expressive fuzzy DL language f$_{KD}$-$\mathcal{SHIN}$ [9]. It allows the user to create a fuzzy knowledge base, based on the description logic Knowledge Representation System Specification (KRSS) which was extended to accommodate the fuzzy elements of fuzzy assertions. The inference services that FiRE supports are: (i) checking consistency of a fuzzy knowledge base, (ii) entailment of fuzzy assertions and (iii) subsumption between two fuzzy concepts. In the following of the paper and in the evaluation procedure we will use the consistency checking inference service.

3 Mapping Validation

Our approach to mapping validation is articulated in four phases

1. *Ontology mapping acquisition.* In this phase, we acquire mappings produced by using an ontology mapping system; the matching system can rely on syntactic, structural or even semantic matching techniques.
2. *Fuzzy interpretation of mappings.* In this phase, the acquired mappings are interpreted as fuzzy assertions as presented in Section 2.1.
3. *Fuzzy reasoning over mappings.* In this phase, the ontology obtained by enriching the second ontology of the mapping with fuzzy individual assertions produced with the help of the mappings is checked for consistency by means of a fuzzy reasoning system.
4. *Mapping validation and revision.* In this phase, mappings are revised according to the reasoning results; mappings causing inconsistencies within the new ontology are refined and given a new strength.

In more detail the validation procedure, takes as input a mapping set (M) together with the respective ontologies (O_1 and O_2) and creates a new mapping set (M'), which includes refined mappings or discarded ones.

The main algorithm is described by **Algorithm-1**. Firstly, M is ordered by descending order. In this way, we first consider the stronger mappings for which similarity is higher. Then, the algorithm examines each mapping with the aforementioned order and calculates a strength. If a mapping was refined

Algorithm 1 $M' :=$ fuzzyValidation(M)

```
input: a mapping set $M$, and the mapped ontologies
output: a validated mapping set $M'$
while the degree of some mapping has changed do
  sort $M$ w.r.t. the strength $n_i$ of each mapping $m_i = \{C_i, C'_i, n_i, R_i\} \in M$
  $M' := \emptyset$
  for $m_i \in M$ do
    $newStrength_i :=$ computeStrength( $m_i$ )
    if $newStrength_i$ is different than $n_i$ then
      $m_i := \{C_i, C'_i, newStrength_i, R_i\}$
      break
    end if
    if $newStrength$ is non zero then
      add $m_i$ to $M'$
    end if
  end for
end while
return $M'$
```

then the same method is applied again for the old set of mappings plus the new refined one, since the new degree might cause a new conflict that did not occur before. This is performed iteratively until all the mappings have been used and no inconsistencies occur. The final set of mappings is saved in M'.

The method that refines the degree of a mapping is described by **Algorithm-2** and proceeds as follows: A new ontology $O' = \langle T', R', A' \rangle$ is created, where $T' = T_2$, $R' = R_2$. The ABox of the new ontology is gradually constructed from the ABox of O_2 and by using the current mapping in order to transfer individuals from ontology O_1. More formally, $A' = A_2 \cup A_M$, where A_M is defined as follows:

$$\begin{aligned} A_M \quad = \quad & \{a : C'_i \geq n \mid \langle C_i, C'_i, n, \sqsubseteq \rangle \in M, O_1 \models C_i(a)\} \cup \\ & \{a : C'_i = n \mid \langle C_i, C'_i, n, = \rangle \in M, O_1 \models C_i(a)\} \cup \\ & \{a : C'_i \leq n \mid \langle C_i, C'_i, n, \sqsupseteq \rangle \in M, O_1 \models C_i(a)\}. \end{aligned}$$

As it can be noted by the above definition, both the explicit as well as inferred assertions are taken into consideration $(O_1 \models C_i(a))$. To do so we make use of a classic DL reasoner and more precisely in the current setting we have used Pellet [10]. For example, if $m_i = \langle C_i, C'_i, 0.8, \equiv \rangle$ and $O_1 \models C_i(a)$ then $A_M = A_M \cup \{a : C'_i = 0.8\}$. After, a new fuzzy individual assertion has been added in O' we call FiRE, in order to check for inconsistencies. If an inconsistency occurs the strength of the mapping is refined, while if an inconsistency does not occur the old degree is retained. The procedure that refines the strength of the mapping is `refineStrength`. This procedure takes as input low level information from the fuzzy reasoner about what conditions created the inconsistency, and according to it proceeds with the refinement of the strength of the mapping so as to restore the consistency in the ontology. For example, a pair of assertions of the form $a : C \geq 0.8$ and $a : C \leq 0.7$ obviously denotes a contradiction.

Algorithm 2 s := computeStrength(m_i)

```
input: $m_i := \{C_i, C'_i, n_i, R_i\}$
output: the new strength of the mapping
for every individual of $C_i$ ($C_i(a)$) do
   add $a$ to $C'_i \longrightarrow C'_i(a)$
   check consistency of $O_2$
   if $O_2$ is not consistent then
      remove all individuals of $C_i$ added to $C'_i$
      $s$ := refineStrength(inconsistencyInfo)
   else
      $s := n_i$
   end if
end for
return $s$
```

Example. Consider two simple ontologies, O_1 and O_2, defined as follows:

$$\begin{aligned} O_1 : &\ MobilePhone \sqsubseteq MobileDevice \\ \\ O_2 : &\ Phone \sqsubseteq ElectronicDevice \\ &\ CablePhone \sqsubseteq Phone \\ &\ CellularPhone \sqsubseteq Phone \\ &\ CablePhone \sqsubseteq \neg CellularPhone \end{aligned}$$

The two ontologies have been compared by adopting the linguistic component of HMatch 2.0 [11], which is based on a combination of terminological and syntactic techniques. The result of the matching process is the following set of mappings:

1. map(*MobileDevice*, *ElectronicDevice*, 0.7)
2. map(*MobilePhone*, *Phone*, 0.6)
3. map(*MobilePhone*, *CablePhone*, 0.4)
4. map(*MobilePhone*, *CellularPhone*, 1.0)

Since the validation process works by translating mappings into fuzzy individual assertions, suppose that each concept of the two ontologies has at least one representative individual. In particular, we assume that mp_1 is an instance of the concept $MobilePhone$ and md_1 is an instance of the concept $MobileDevice$. Sorted by the strength, one by one mappings are inserted into the second ontology as fuzzy individual assertions.

Following the example, the first mapping to be added to O_2 is mapping 4, which is translated into the assertion ($CellularPhone(mp_1)$, 1.0). Since the first mapping does not cause an inconsistency, the procedure moves to the subsequent mapping (1), which is converted into (*ElectronicDevice*(md_1), 0.7) and (*ElectronicDevice*(mp_1), 0.7). The latter assertion violates the fuzzy DLs interpretation of subsumption ($C \sqsubseteq D \Longleftrightarrow C^{\mathcal{I}}(a) \leq D^{\mathcal{I}}(a)$), therefore making the resulting ontology inconsistent. In this case, the solution is to increase the strength of $ElectronicDevice(mp_1)$ and $ElectronicDevice(md_1)$ to 1 in order to satisfy the semantic constraint $ElectronicDevice^{\mathcal{I}}(mp_1^{\mathcal{I}}) \geq CellularPhone^{\mathcal{I}}(mp_1^{\mathcal{I}})$. The same situation occurs when the assertions corresponding to mapping 2 are added into O_2 and the same refinement is applied to restore consistency. At last, the assertion determined by mapping 3, i.e. $(CablePhone(mp_1), 0.4)$, causes an inconsistency because it does not satisfy the semantic constraint $CablePhone^{\mathcal{I}}(mp_1^{\mathcal{I}}) \leq 1{-}CellularPhone^{\mathcal{I}}(mp_1^{\mathcal{I}})$. Giving priority to the stronger mapping, the latest assertion has to be refined. Since the resulting strength would be equal to 0, the assertion corresponding to mapping 3 is definitely dropped, and the mapping is removed as well. The result of the validation process is the following mapping set:

1. map($MobileDevice$, $ElectronicDevice$, 1.0)
2. map($MobilePhone$, $Phone$, 1.0)
3. map($MobilePhone$, $CellularPhone$, 1.0)

4 Conflict Resolution

The validation process described in the previous section enforces the inconsistency detection and resolution by refining the strength of the mappings. When a conflict arises, two or more mappings are involved and, to achieve the consistency, at least one of them must be refined or removed. Generally the choice among the conflicting mappings is not trivial because it should be driven by the semantics of the mapped elements. The decision is even a harder task when is performed automatically, therefore requiring effective heuristics. Moreover, even when the choice is made by a human expert, there can be different correct decisions according to different criteria that can be adopted.

The proposed validation technique adopts a naive strategy which gives priority to the strongest mapping and forces the last added mapping to be refined or deleted. This solution has the advantage of being efficient in terms of performances but does not always lead to the expected results. In fact, for instance, one may prefer to preserve the highest number of mappings instead of the strongest ones. The limitation is more evident if we consider mapping deletion as the only possible way to solve inconsistencies. For instance, consider the two ontologies defined in the example of the previous section and assume to have the same mapping set but with the following strength values:

1. map($MobileDevice$, $ElectronicDevice$, 0.5)
2. map($MobilePhone$, $Phone$, 0.7)
3. map($MobilePhone$, $CablePhone$, 0.8)
4. map($MobilePhone$, $CellularPhone$, 0.6)

The conflicting subsets of mappings in this configuration are (1,2,3) and (3,4), due to the violation of the fuzzy DLs interpretation of subsumption and negation, respectively. If we apply a restricted version of the validation procedure of Section 3 that allows only the deletion of inconsistent mappings, the inconsistency would be solved by deleting all the mappings except for mapping 3, which is the strongest one. In this case, it is clear that giving priority to the mapping with the highest value could be not always the expected choice.

To provide a better support for the resolution of mapping inconsistencies, we propose a different approach, namely the *conflict resolution method*, based on the complete analysis of the conflicts. The underlying idea is to compute a *degree of inconsistency* of each mapping, i.e. a measure that reflects the number of times in which a mapping is involved in a conflict. To evaluate this degree, we consider the inconsistencies in all the possible mapping configurations, that are the set $\overline{\mathcal{P}(M)}$ of all the subsets of the given mapping set, except the empty set and the singleton sets. More formally, given a set of mappings M and the set $\overline{\mathcal{P}(M)} \equiv \mathcal{P}(M) \setminus \{x \in \mathcal{P}(M) \mid x = \emptyset \vee |x| = 1\}$, we define the conflicting set $\mathcal{C}(M) \subseteq \overline{\mathcal{P}(M)}$ as

$$\mathcal{C}(M) = \{c \in \overline{\mathcal{P}(M)} \mid \exists\, m, m' \in c \text{ such that } m \text{ and } m' \text{ cause an inconsistency}\}$$

$\mathcal{C}(M)$ is built by validating each subset $s_i \in \overline{\mathcal{P}(M)}$ through the validation procedure of Section 3. If the resulting set s'_i is equal to s_i then s_i does not contain any conflict and it is not included into $\mathcal{C}(M)$. Otherwise, if $s'_i \subset s_i$ then a mapping has been removed to solve an inconsistency, therefore s_i is added into $\mathcal{C}(M)$.

We define the minimal conflicting set $\mathcal{MC}(M)$ of M as the collection of all minimal subset of mappings which contains a conflict:

$$\mathcal{MC}(M) = \{mc \in \mathcal{C}(M) \mid \nexists\, mc' \in \mathcal{C}(M) \text{ such that } mc' \subseteq mc\}$$

The degree of inconsistency i_m of a mapping $m \in M$ is defined as follows:

$$i_m = |\{mc \in \mathcal{MC}(M) \mid m \in mc\}|$$

The assumption is that the higher is the degree of inconsistency of a mapping, the more benefit we will get by removing it from the mapping set. Therefore, the strategy behind this conflict resolution method is to preserve as much as possible the mappings by detecting and deleting those which participate in the highest number of conflicts. After computing the degree of inconsistency, all the mappings are added into the second ontology as fuzzy individual assertions and the resulting ontology is checked for consistency. If an inconsistency is detected, the mapping with the highest degree of inconsistency is removed and the resulting ontology is again checked for consistency. The step is repeated until consistency is achieved.

Let us describe this method with the aforementioned set of mappings that are not correctly validated by the strength-based ordering approach. The computation of the degrees of inconsistency produces the following results:

$$\begin{aligned}
\overline{\mathcal{P}(M)} &= \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4),(1,2,3),(1,2,4),\ldots\}\\
\mathcal{C}(M) &= \{(1,2),(1,3),(3,4),(1,2,3),(1,2,4),(2,3,4),(1,2,4),(1,2,3,4)\}\\
\mathcal{MC}(M) &= \{(1,2),(1,3)(3,4)\}
\end{aligned}$$

$$i_1 = |\{(1,2),(1,3)\}| = 2, \quad i_2 = |\{(1,2)\}| = 1, \quad i_3 = 2, \quad i_4 = 1$$

All mappings are added into the resulting ontology and, subsequently, mappings 1 and 3 are removed before consistency is restored. Compared with the results of the strength-based ordering approach, this method detected the actual incorrect mapping (3) and produced the configuration with the largest number of mappings. Moreover, in the context of semi-automatic validation tools, the analysis performed with this method can report to the user the actual minimal sets of conflicting mappings, in order to better support the decision process.

5 Related Work

Recent work [12] have focused on mapping validation as a post-processing task over mappings produced by other matchmaking tools. Grounded on the theories of the Distributed Description Logics, the process consists in translating mappings into bridge rules (i.e. inter-ontology semantic relations) and check for the consistency of the resulting distributed knowledge base. The approach does not handle the inherent uncertainty of mapping caused by the possible inaccuracy of the heuristics adopted by the matching techniques.

As a possible solution to cope with the uncertainty of automatically discovered mappings, probabilistic techniques have been developed. The approach presented in [13] translates the mapped ontologies into bayesian networks and treats concept mapping between the two ontologies as evidential reasoning between the two translated BN. In our foregoing work on mapping validation [2], starting from the crisp approach in [12], we refined the validation process by attaching to mappings a probability measure determined by the confidence value of the mapping. The probability value is interpreted as the likelihood of the mapping being correct. The resulting relations are interpreted according to the probabilistic description logics, which provides consistency check and inference services in order to perform validation. A similar approach has been presented in [1], where the combination of a rule-based framework and Probabilistic Description Logic Programs is exploited to validate and merge mappings produced by different techniques and tools. As in [2], the confidence value is interpreted as a probability measure of the mapping correctness.

To be effective, probabilistic approaches should be fed with values which actually state the confidence of the relation, therefore computed on the basis of well-founded statistical techniques or measures. This turns out to be a relevant limitation because most of the matchmaking tools do not provide such a measure but only a value representing the degree of similarity between the mapped elements. The alternative we propose is to exploit the fuzzy interpretation to handle the uncertainty of mappings but without relying on the confidence values. In the ontology matching literature, fuzzy theories have been exploited mainly with the aim of dealing with uncertainty during the process of mapping discovery and not for validation. For instance, the method described in [14] formulates the ontology mapping problem as a rule application problem in the fuzzy conceptual graph model. In our approach, based on the fuzzy description logics, the numeric value attached to a mapping is intended as a degree of truth of the relation. According to the way the numeric value is computed in most of the matching techniques, the fuzzy interpretation is more suitable compared to the probability value, especially when mappings represent a generic similarity relation between the concepts.

Regarding conflict resolution strategies, relevant work have been presented in the field of ontology repairing in order to provide debugging functionalities for logically erroneous knowledge bases. In [15], minimal incoherence-preserving sub-TBoxes (MIPS) are defined as the smallest subsets of an original TBox preserving unsatisfiability of at least one atomic concept. MIPS are detected and solved through a tableaux-like technique. Our definition of the degree of inconsistency adopts the same principle but applied to the mapping conflict resolution problem.

Other work in dealing with ontology mapping in the fuzzy context has been presented in [16] where Li et al. have introduced E-Connections integrated into extended fuzzy description Logics (EFDLs) that couple both fuzzy and distributed features within description logics and in [17], where Lu et al. propose a discrete tableau algorithm to achieve reasoning within the logical system of EFDLs. Unfortunately, not practical implementation of the algorithm is known, in order to be used in a practical setting for reasoning over such fuzzy mappings.

6 Concluding Remarks

In this paper we have discussed the application of the fuzzy DLs theories to the problem of mapping validation as a different way of handling mapping uncertainty with respect to probabilistic approaches. As a result, we described a mapping validation algorithm based on fuzzy interpretation of mappings in order to detect inconsistencies. Similarly to previous work on mapping validation, the strategy to solve inconsistencies is a simple strength-based heuristics, i.e. the conflicting mapping with the highest strength value is preserved. Although being a fast solution, this naive approach does not lead always to the expected configuration. To cope with possible different strategies, we proposed a conflict resolution approach which performs a thorough analysis of all possible inconsistencies and computes the minimal sets of conflicting mappings.

The preliminary results show that the conflict resolution method is effective and can potentially be applied to any validation semantics (e.g. probabilistic, fuzzy). Furthermore, other validation strategies can be built on top of it, for instance a strategy to maximize the number of preserved mappings. Regarding the complexity, it is obviously dependent on the number of mappings involved and, without further optimizations, the method is applicable only on relatively small alignments. Future work will be devoted to the development of optimization techniques, in particular the goal is to reduce the number of mapping subsets to be validated during the search for the minimal conflicting sets. A possible way of reducing the search space, and thus the combinatorial space, is to make some approximations, like the one proposed in [18]. Moreover, the proposed validation procedure supports only subsumption and equivalence, therefore further investigation are needed to include other kind of correspondences between aligned ontology elements.

References

1. Calì, A., Lukasiewicz, T., Predoiu, L., Stuckenschmidt, H.: A framework for representing ontology mappings under probabilities and inconsistency. In: URSW. (2007)
2. Castano, S., Ferrara, A., Lorusso, D., Nth, T.H., Moeller, R.: Mapping validation by probabilistic reasoning. In: Proceedings of the 5th European Semantic Web Conference. LNCS, Springer Verlag (2008)
3. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
4. Straccia, U.: Towards a fuzzy description logic for the semantic web. In: Proceedings of the 2nd European Semantic Web Conference. (2005)
5. Stoilos, G., Stamou, G., Pan, J.: Handling imprecise knowledge with fuzzy description logics. In: Proceedings of the International Workshop on Description Logics (DL 2006), Lake District, UK. (2006)
6. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. Web Semantics **1** (2003) 7–26
7. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. In Spaccapietra, S., March, S.T., Aberer, K., eds.: J. Data Semantics I. Volume 1 of Lecture Notes in Computer Science., Springer (2003) 153–184
8. Simou, N.: Fuzzy Reasoning Engine FiRE, (http://www.image.ece.ntua.gr/ nsimou/FiRE/)
9. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J.Z., Horrocks, I.: Reasoning with very expressive fuzzy description logics. Journal of Artificial Intelligence Research **30** (2007) 273–320
10. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Journal of Web Semantics **5** (2007) 51–53
11. Castano, S., Ferrara, A., Montanelli, S.: Matching Ontologies in Open Networked Systems: Techniques and Applications. Journal on Data Semantics (JoDS) **V** (2006)
12. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing ontology mappings. In: AAAI. (2007) 1408–1413
13. Pan, R., Ding, Z., Yu, Y., Peng, Y.: A Bayesian Network Approach to Ontology Mapping. In: Proceedings of the Fourth International Semantic Web Conference. (2005)
14. Buche, P., Dibie-Barthélemy, J., Ibanescu, L.: Ontology mapping using fuzzy conceptual graphs and rules. In: ICCS Supplement. (2008) 17–24
15. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI, Morgan Kaufmann (2003) 355–362
16. Li, Y., Xu, B., Lu, J., Kang, D.: A distributed and fuzzy extension of description logics. In: KES (1). (2006) 655–662
17. Lu, J., Li, Y., Zhou, B., Kang, D., Zhang, Y.: Distributed reasoning with fuzzy description logics. In: International Conference on Computational Science (1). (2007) 196–203
18. Straccia, U., Troncy, R.: oMAP: Combining classifiers for aligning automatically OWL ontologies. In: 6th International Conference on Web Information Systems Engineering (WISE-05). Number 3806, Springer Verlag (2005) 133–147

On fixing semantic alignment evaluation measures

Jérôme David and Jérôme Euzenat

LIG & INRIA Grenoble Rhône-Alpes
Grenoble, France
Jerome.{David|Euzenat}@inrialpes.fr

Abstract. The evaluation of ontology matching algorithms mainly consists of comparing a produced alignment with a reference one. Usually, this evaluation relies on the classical precision and recall measures. This evaluation model is not satisfactory since it does not take into account neither the closeness of correspondances, nor the semantics of alignments. A first solution consists of generalizing the precision and recall measures in order to solve the problem of rigidity of classical model. Another solution aims at taking advantage of the semantic of alignments in the evaluation. In this paper, we show and analyze the limits of these evaluation models. Given that measures values depend on the syntactic form of the alignment, we first propose an normalization of alignment. Then, we propose two new sets of evaluation measures. The first one is a semantic extension of relaxed precision and recall. The second one consists of bounding the alignment space to make ideal semantic precision and recall applicable.

1 Introduction

With the semantic Web, many related but heterogenous ontologies are being created. In such an open context, there is no reason why two domain-related applications would share the same ontologies. In order to facilitate the exchange of knowledge between such applications, ontology matching aims at discovering a set of relations between entities from two ontologies. This set of relations is called an alignment.

Many different matching algorithms have been designed [Euzenat and Shvaiko, 2007]. In order to compare the performance of such algorithms, some efforts are devoted to the evaluation of ontology matching tools. Since 2004, the Ontology Alignment Evaluation Initiative[1] (OAEI) organizes, every year, an evaluation of ontology matching methods. The evaluation of matching algorithms consists of comparing a produced alignment with a reference one. This evaluation often relies on two classical measures used in information retrieval: precision and recall [van Rijsbergen, 1979]. Precision measures the ratio of correct correspondences in the evaluated alignment. Recall measures the ratio of reference correspondence found by the evaluated alignment.

In the context of alignment evaluation, precision and recall present the drawbacks to be all-or-nothing measures [Ehrig and Euzenat, 2005] and they do not consider neither the semantic of alignment relations, nor those of ontologies. Then, an alignment can be very close to the expected result and have low precision and recall values. Two approaches have been proposed for correcting these drawbacks. [Ehrig and Euzenat,

[1] http://oaei.ontologymatching.org

2005] introduced a generalization of precision and recall measures. This approach relies on syntactic measures relaxing the all-or-nothing feature of classical measures in order to take into account close correspondences. [Euzenat, 2007] has introduced semantic precision and recall measures which rely on a semantic of alignments

We will show that semantic precision and recall are still dependent on the alignment syntax and, as a consequence, they can assign different values to semantically equivalent alignments. [David, 2007] proposes to use the ideal semantic precision and recall measures introduced in [Euzenat, 2007] restricted to alignments containing only simple correspondences, i.e. only between named entities.

In this paper, we show and analyze the limits and problems of the semantic precision and recall measures. To overcome their drawbacks, we first investigate an approach allowing to normalize alignments. This normalization relies on algebra of alignment relations and can partially resolves problems encountered by the evaluation measures. In addition, we propose two adaptations of the relaxed and semantic measures. The first adaptation makes use of the generalization framework of [Ehrig and Euzenat, 2005] and allows to locally consider the semantic of alignments. The second one is a restriction of Semantic closures of alignment. This restriction makes the ideal semantic measures proposed in [Euzenat, 2007] useable.

This paper is organized as follows: a first section introduces the definitions related to the syntax and semantics of alignments. In the second section, we first present and introduce five properties that an ideal model should satisfy. Then, we present the classical evaluation measures and the semantic evaluation measures which satisfy three of the five desired properties. In the following section, we explain why these semantic measures do not satisfy the two last properties. The last section proposes three ways for fixing the semantic measures: a normalization of alignments, new relaxed semantic precision and recall measures, and Λ-bounded semantic evaluation measures.

2 Ontology alignment: syntax and semantic

2.1 Definition and syntax

An alignment groups correspondences between entities or formulas from two ontologies o_1 and o_2. Each element of correspondence can be associated to a quality value by a function q. We use the following syntax for representing an alignment:

Definition 1 (Alignment). *An **alignment** between two ontologies o_1 and o_2 is a set of correspondances holding between o_1 and o_2. A correspondance, noted $c = (x, y, \mathcal{R})$, is a triple where x, respectively y, are formulas (or entities) from o_1, respectively from o_2, and $\mathcal{R}$ is the relation holding between x and y. A correspondance $c = (x, y, \mathcal{R})$ can be also written $x\mathcal{R}y$*

This definition includes both simple alignments considering only matching relations between entities (classes or properties) and complex alignments containing relations between formula inferred from the ontologies.

2.2 Semantic of alignments: Semantic closure and semantic reduction

Alignments between ontologies can be helpful for reasoning with several ontologies. For enabling reasoning capabilities, a semantic for alignments must be defined. In this paper, we relies on the semantic proposed in [Euzenat, 2007]. This semantic of alignment is function of the semantics of each individual ontology. The semantic of an ontology is given by its set of models.

Definition 2 (Model). *a model $m = \langle I, D \rangle$ of o is a function I from the terms of o to a domain of interpretation D, which satisfies all the assertions in o:*

$$\forall \delta \in o, m \models \delta$$

The set of models of an ontology o is denoted as $\mathcal{M}(o)$.

Because the models of various ontologies can have different interpretation domains, we use the notion of an equalising function, which helps make these domains commensurate.

Definition 3 (Equilising function). *Given a family of interpretations $\langle I_o, D_o \rangle_{o \in \Omega}$ of a set of ontologies Ω, an equalising function for $\langle I_o, D_o \rangle_{o \in \Omega}$ is a family of functions $\gamma = (\gamma_o : D_o \longrightarrow U)_{o \in \Omega}$ from the ontology domains of interpretation to a global domain of interpretation U. The set of all equalising functions is called Γ.*

The relations used in correspondences do not necessarily belong to the ontology languages. As a consequence, a semantics for them must be provided.

Definition 4 (Interpretation of alignment relations). *Given $\mathcal{R}$ an alignment relation and U a global domain of interpretation, $\mathcal{R}$ is interpreted as a binary relation over U, i.e., $\mathcal{R}^U \subseteq U \times U$.*

The definition of correspondence satisfiability relies on γ and the interpretation of relations. It requires that in the equalised models, the correspondences are satisfied.

Definition 5 (Satisfied correspondence). *A correspondence $c = \langle x, y, \mathcal{R} \rangle$ is satisfied for an equalising function γ by two models m, m' of o, o' if and only if $\gamma_o \cdot m \in \mathcal{M}(o)$, $\gamma_{o'} \cdot m' \in \mathcal{M}(o')$ and*

$$\langle \gamma_o(m(e)), \gamma_{o'}(m'(e')) \rangle \in \mathcal{R}^U$$

This is denoted as $m, m' \models_\gamma c$.

Given an alignment between two ontologies, the semantics of the aligned ontologies can be defined as follows.

Definition 6 (Models of aligned ontologies). *Given two ontologies o and o' and an alignment A between these ontologies, a model m'' of these ontologies aligned by A is a triple $\langle m, m', \gamma \rangle \in \mathcal{M}(o) \times \mathcal{M}(o') \times \Gamma$, such that $m, m' \models_\gamma A$.*

We will consider a specific kind of consequence, α-consequences [Euzenat, 2007], which are the correspondences holding for all models of aligned ontologies.

Definition 7 **(α-Consequence of aligned ontologies).** *Given two ontologies o and o' and an alignment A between these ontologies, a correspondence δ is a α-consequence of o, o' and A (noted $A \models \delta$) if and only if for all models $\langle m, m', \gamma \rangle$ of o, o' and A, $m, m' \models_\gamma \delta$ (the set of α-consequences is noted by $Cn(A)$).*

Given this semantic, the semantic closure and semantic reduction of an alignment are given by the following definitions:

Definition 8 **(Semantic closure).** *The semantic closure $Cn(A)$ of an alignment A is the set of its α-consequences.*

Obviously , the semantic closure of an alignment is unique but it has no reason to be finite.

Definition 9 **(Semantic reduction).** *A semantic reduction (or minimal cover) A^0 of an alignment A is an alignment satisfying $Cn(A^0) = Cn(A)$ and $\forall c \in A^0,\ Cn(A^0 - \{c\}) \neq Cn(A)$*

There could exist several semantic reductions for a given alignment. An alignment A contains redundant elements if A is not a minimal cover. A correspondence $c \in A$ is redundant if $A - \{c\} \models c$.

3 Evaluation models

Alignment evaluation is achieved by comparing the produced alignment with the reference one. This comparison usually relies on the precision (P) and the recall (R) measures [van Rijsbergen, 1979]. Intuitively, the precision aims at measuring the correctness of the evaluated alignment. The recall is used for quantifying the completeness of the evaluated alignment.

In the rest of this paper, we will consider two alignments between ontologies o_1 and o_2: a reference alignment, noted A_r, and an alignment produced by some matching method A_e.

3.1 Desired properties of evaluation measures

If we consider that precision and recall should approximate correctness and completeness, an ideal model taking semantic into account, would respect the constraints given by [Euzenat, 2007]:

- $A_r \models A_e \Rightarrow P(A_e, A_r) = 1$ (max-correctness)
- $A_e \models A_r \Rightarrow R(A_e, A_r) = 1$ (max-completeness)
- $Cn(A_e) = Cn(A_r)$ iff $P(A_e, A_r) = 1$ and $R(A_e, A_r) = 1$ (definiteness)

Furthermore, in the evaluation context, one could be interested to compare several alignments produced by some matching algorithms against one reference alignment. Then, it would be useful that two semantically equivalent alignments have the same precision and recall values.

- $Cn(A_{e_1}) = Cn(A_{e_2}) \Rightarrow P(A_{e_1}, A_r) = P(A_{e_2}, A_r)$ and $R(A_{e_1}, A_r) = R(A_{e_2}, A_r)$ (semantic-equality)

Finally, if the evaluated and the reference alignments share some common information then the precision and recall values must not be null:

- $P(A_e, A_r) = 0$ and $R(A_e, A_r) = 0$ iff $Cn(A_e) \cap Cn(A_r) = Cn(\emptyset)$ (overlapping positiveness)

3.2 Classical evaluation model

The classical evaluation model is based on the interpretation of alignments as sets and considers the following sets :

- ***E***: the set of all correspondences that could be generated between o_1 and o_2. This set is a subset of the cartesian product of all entities which can be deduced from o_1, those deductible for o_2 and the set of matching relations considered.
- **true-positives**: the set of correspondences which are found by the matching method and contained in the reference alignment.
- **false-positives**: the set of correspondences which are found by the matching method but not contained in the reference alignment.
- **false-negatives**: the set of reference correspondences which are not found by the matching method.
- **true-negatives**: the set of correspondences that are neither in the evaluated alignment nor in the reference alignment.

	relevant	not relevant	
found	$\vert A_e \cap A_r \vert$ true-positives	$\vert A_e - A_r \vert$ false-positives	$\vert A_e \vert$
not found	$\vert A_r - A_e \vert$ false-negatives	$\vert (E - A_r) - A_e \vert$ true-negatives	$\vert E - A_e \vert$
	$\vert A_r \vert$	$\vert E - A_r \vert$	

Table 1. Contingency of sets A_e and A_r.

The cardinalities of these sets are given in the contingency table 1. The sets of true-positives, false negatives, and false-positives are defined only from A_e and A_r. The set of true-negatives is also function of the set E which is not easily identifiable.

From these contingencies, the classical measure of precision and recall can be defined. The precision (P) represents the proportion of found correspondences that are relevant:

$$P(A_e, A_r) = \frac{|A_e \cap A_r|}{|A_e|} \tag{1}$$

The recall (R) represents the proportion of relevant correspondences that have been found :

$$R(A_e, A_r) = \frac{|A_e \cap A_r|}{|A_r|} \tag{2}$$

3.3 Limitations of classical precision and recall

These two measures applied to this simple model have the advantages to be easily computable and understandable. However, they verify none of the constraints presented Section 3.1. This is because they do not consider the semantic of alignment relations, nor the semantic of ontologies.

Firstly, they do not take into account the semantic of matching relations. For example, if the produced alignment A_e contains the elements $x \sqsubseteq y$ and $x \sqsupseteq y$, and the reference alignment A_r contains the element $x \equiv y$, then the classical model will consider $x \sqsubseteq y$ and $x \sqsupseteq y$ as false-positives and $x \equiv y$ as false-negative. In this case the precision and recall values are equals to 0 even if $A_e \equiv A_r$.

Secondly, this classical model does not take the semantic of ontologies into account. For example, A_e contains the element $x' \sqsubseteq y$, the reference alignment A_r contains the element $x \equiv y$, and the ontology o_1 states $x' \sqsubseteq x$. Even if $A_r \models A_e$, the classical precision will be equal to 0 since the correspondence $x' \sqsubseteq y$ is considered as a false-positive by this evaluation model.

3.4 Semantic evaluation models

In order to resolve the drawbacks of classical precision and recall, [Euzenat, 2007] proposes to take into account the semantics of matching relations and ontologies. The author provides two extensions of precision and recall.

The ideal extension of the classical model consists of replacing A_e and A_r by their respective sets of α-consequences, $Cn(A_e)$ and $Cn(A_r)$. Table 2 show the new contingencies.

	relevant	not relevant	
found	$\lvert Cn(A_e) \cap Cn(A_r)\rvert$ true-positives	$\lvert Cn(A_e) - Cn(A_r)\rvert$ false-positives	$\lvert Cn(A_e)\rvert$
not found	$\lvert Cn(A_r) - Cn(A_e)\rvert$ false-negatives	$\lvert (E - Cn(A_r)) - Cn(A_e)\rvert$ true-negatives	$\lvert E - Cn(A_e)\rvert$
	$\lvert cn(A_r)\rvert$	$\lvert E - Cn(A_r)\rvert$	

Table 2. Contingencies of the ideal extension of the classical model.

From this extended model, ideal precision and recall measures, respectively named P_i and R_i, are :

$$P_i(A_e, A_r) = \frac{|Cn(A_e) \cap Cn(A_r)|}{|Cn(A_e)|} \tag{3}$$

$$R_i(A_e, A_r) = \frac{|Cn(A_e) \cap Cn(A_r)|}{|Cn(A_r)|} \tag{4}$$

These measures correct the drawbacks of the classical model and all the properties given Section 3.1 are satisfied. However, they bring a new problem : as the semantic closures of alignments could be infinite, then the measures may be undefined.

In order to overcome this problem, [Euzenat, 2007] introduces two new measures known as semantic precision and semantic recall.

Semantic precision measures the proportion of evaluated correspondances of A_e that can be deduced from A_r.

$$P_s(A_e, A_r) = \frac{|A_e \cap Cn(A_r)|}{|A_e|} \tag{5}$$

Semantic recall measures the proportion of reference correspondances of A_r that can be deduced from A_e.

$$R_s(A_e, A_r) = \frac{|Cn(A_e) \cap A_r|}{|A_r|} \tag{6}$$

With these measures, the max-correctness, max-completeness, and definiteness properties are preserved. The values of semantic precision and semantic recall are greater than or equal to those of classical ones because $|A_e \cap Cn(A_r)| > |A_e \cap A_r|$ and $|Cn(A_e) \cap A_r| > |A_e \cap A_r|$.

4 Limitations of semantic precision and recall

Semantic precision and recall correct some drawback of classical precision and recall measure since they satisfy the max-correctness, max-completeness, and definiteness properties. Nevertheless, they do not satisfy the semantic-equality and overlapping-positiveness properties which we have introduced. This is due to the fact that these semantic measures are still dependent on the syntactic form of the alignments.

4.1 Limitation concerning semantic-equality property

Two alignments A_{e_1} and A_{e_2} having the same closure and then, semantically equivalent, could have different precision and recall values according to A_r. This due to the fact that the semantic precision and recall are directly function of the cardinalities of the correspondences sets which could be different for two semantically equivalent alignments.

We give two examples demonstrating that semantic evaluation measures do not satisfy the semantic-equality property. In the first example, we reason only with alignment. In the second example, we show that redundancy in alignment can break the satisfaction of semantic-equality property by precision measure.

In the first example, we consider two alignments $A_{e_1} = \{x \equiv y, u \equiv v\}$ and $A_{e_2} = \{x \sqsubseteq y, x \sqsupseteq y, u \equiv v\}$. These two alignments are equivalent since we have only replaced the equivalence $x \equiv y$ of A_{e_1} by $x \sqsubseteq y$ and $x \sqsupseteq y$ in A_{e_2}. According to a reference alignment $A_r = \{x \equiv y\}$, the two alignments do not have the same precision values: $P_s(A_{e_1}, A_r) = 1/2$ and $P_s(A_{e_2}, A_r) = 2/3$.

In the second example, we now have the alignments $A_{e_1} = \{x \equiv y, u \equiv v\}$ and $A_{e_2} = \{x' \sqsubseteq y, x \equiv y, u \equiv v\}$, and the knowledge $o_1 \models x' \sqsubseteq x$. These two alignments are equivalent since $x' \sqsubseteq y$ is redundant according to $x \equiv y$. Nevertheless, the semantic precision values will be different: $P_s(A_{e_1}, A_r) = 1/2$ and $P_s(A_{e_2}, A_r) = 2/3$

4.2 Limitation concerning overlapping-positiveness property

An alignment could have a null precision or/and recall value even if the intersection of its consequence sets and those of the reference is not the empty set. This due to the fact that the semantic precision and recall partially take the alignment semantic into account. A correspondance can entail several correspondances. Such a correspondance can be partially true-positive in the sense that it entails a true-positive element but also a false-negative or false-positive element. With the semantic precision and recall, such elements are entirely considered as false-positives or/and false-negatives.

For example, let be the two alignments $A_e = \{a\}$ and $A_r = \{b\}$, another matching relation c and the properties $a \models c$, $b \models c$, $a \not\models b$ and $b \not\models a$. On this trivial example, the semantic precision and recall values are both equals to 0 even if the intersection of their Semantic closures is not equals to the empty set (i.e. $c \in Cn(A_e) \cap Cn(A_r)$).

5 Corrections of semantic evaluation measures

In previous section, we highlighted some drawbacks of classical precision and recall and semantic precision and recall. The first kind of problems concerns the inability of classical and generalized precision and recall measures to reason with the alignment relations. The semantic precision and recall try to resolve this problem by using Semantic closures, but these measures are still defined on the alignment cardinality which is dependent on the syntactic form of the alignment. As a consequence, there are some cases where the semantic-equality property is not satisfied.

When this problem is entirely due to the syntactic form the alignments, we may try to resolve it by normalizing the alignment representation. We propose here a normalization strategy which relies on algebras of alignment relations [Euzenat, 2008].

Then, with the help of alignment normalization, we propose two new sets of evaluation measures. The first one concerns relaxed semantic measures based on the generalized precision and recall framework of [Ehrig and Euzenat, 2005]. Contrarily to the original generalized precision and recall measures provided in the aforementioned paper, these new measures are not only based on the syntactic form of alignment, but also on the semantic of alignments.

The second set of measures is an adaptation of ideal semantic measures of [Euzenat, 2007].

5.1 Normalization of alignments

For allowing measures to respect the semantic-equality property, it is useful to introduce a notion of a normal form for alignments. A normal form for alignments ensures that two semantically equivalent alignments have always the same syntax or form. Naturally, it is a very difficult problem but we can propose a partial solution which only considers the semantic of alignment relations. Our notion of normal form takes benefit of entailment capabilities provided by algebras of alignment relations and does not use any knowledge about the aligned ontologies.

An algebra of alignment relations [Euzenat, 2008] is a particular type of relation algebra [Tarski, 1941] defined by the tuple $\langle 2^{\Gamma}, \cap, \cup, \cdot, \Gamma, \emptyset, \{\equiv\}, ^{-1} \rangle$ where Γ is the set of all elementary relations; $\cap$ and $\cup$ are set-operations used to meet and join two sets of relations, for example, if $x\mathcal{R}y$ or $x\mathcal{R}'y$ then $x\mathcal{R} \cup \mathcal{R}'y$; $\cdot$ is the composition operator, i.e. an associative internal composition law with $\{\equiv\}$ as unity element; $^{-1}$ the converse operator. For instance, if $\Gamma = \{\sqsubset, \sqsupset, \equiv, \between, \perp\}$, all all elementary relations, except $\sqsubset$ and $\sqsupset$, are there own converse and, $\sqsubset^{-1}$=$\sqsupset$ and $\sqsupset^{-1}$=$\sqsubset$.

Such an algebra allows to write any relation between entities (or formulas) as a disjunction of elementary relations. For example, $x \sqsubseteq y$ would be written $x\{\sqsubset, \equiv\}y$. With the help of this relation algebra, any pair of entities or formulas will appear at most once in the alignment.

Definition 10. *An* ***alignment in normal form*** *is an alignment $A = (V, q)$ where the set of correspondances V satisfies the following properties:*

1. *$V \subset \{x\mathcal{R}y | x \in o_1 \wedge y \in o_2 \wedge \ \mathcal{R} \subseteq \Gamma\}$: all relations between two entities (or formulas) are written with a disjunction of elementary relations.*
2. *$\forall x\mathcal{R}y \in V, \ \nexists x\mathcal{R}'y \in V, \ \mathcal{R} = \mathcal{R}'$: any pair of entities (or formulas) appear at most once in the alignment.*

Using such a normalization allows to correct classical and semantic precision and recall when relations between entities or formulas are split into several correspondances. For example, let be $A_e = \{x \sqsubseteq y, x \sqsupseteq y\}$ and $A_r = \{x \equiv y\}$. By rewriting these alignments using disjunction of elementary relations, $A_e = \{x\{\sqsubset, \equiv\} \cap \{\sqsupset, \equiv\}y\} = \{x\{\equiv\}y\}$ and $A_r = \{x\{\equiv\}y\}$ will be syntactically equivalent.

Of course, when this problem is due to the semantic of ontologies such a normalization is not sufficient. For example, let be $A_e = \{x \sqsubseteq y\}$ and $A_r = \{x \sqsubseteq z\}$ and the axiom $y \equiv z \in o_2$. These alignments are equivalent (given the previous axiom), but their normalization ($A_e = \{x\{\sqsubset, \equiv\}y\}$ and $A_r = \{x\{\sqsubset, \equiv\}z\}$) are not equal.

5.2 Relaxed semantic precision and recall

In generalized precision and recall framework, evaluation measures are function of a measure quantifying the proximity between two correspondences [Ehrig and Euzenat, 2005]. We propose new proximity measures σ dealing partially with the semantic of alignments. We want such measures to locally respect the max-correctness and max-completeness properties contrarily to those provided in [Ehrig and Euzenat, 2005]:

- if $x'\mathcal{R}'y' \models x\mathcal{R}y$ then $\sigma_{prec}(x\mathcal{R}y, x'\mathcal{R}'y') = 1$ (local max-correctness)
- if $x\mathcal{R}y \models x'\mathcal{R}'y'$ then $\sigma_{rec}(x\mathcal{R}y, x'\mathcal{R}'y') = 1$ (local max-completeness)

In order to propose such measures, we suggest to take advantage of an algebra of alignment relations as presented in the previous section (Section 5.1). Following the example of the relaxed precision and recall, which are oriented, we introduce two new σ measures: σ_{prec} for precision and σ_{rec} for the recall. In these two σ measures, we do not consider the confidence values.

In a first instance, we only consider the case where we have two correspondances aligning the same entities or formulas. Let $x\mathcal{R}y$ and $x\mathcal{R}'y$ be two such correspondances. From the algebra of alignment relations, we have the following properties:

– if $x\mathcal{R}'y \models x\mathcal{R}y$, then $\mathcal{R}' \subseteq \mathcal{R}$,
– if $x\mathcal{R}y \models x\mathcal{R}'y$, then $\mathcal{R} \subseteq \mathcal{R}'$.

Hence, σ_{prec} and σ_{rec} are defined by:

$$\sigma_{prec}(\mathcal{R}, \mathcal{R}') = \frac{|\mathcal{R} \cap \mathcal{R}'|}{|\mathcal{R}'|} \tag{7}$$

$$\sigma_{rec}(\mathcal{R}, \mathcal{R}') = \frac{|\mathcal{R} \cap \mathcal{R}'|}{|\mathcal{R}|} \tag{8}$$

Now, for extending these measures to correspondances which do not align the same entities or formulas, we propose to use this relation algebra also with the ontologies.

Definition 11 (Relaxed semantic proximity measures). *Given an evaluated relation $x\mathcal{R}y$, a reference relation $x'\mathcal{R}'y'$, and relations deduced from ontologies, $o_1 \models x\mathcal{R}_1x'$ and $o_2 \models y\mathcal{R}_2y'$, the relaxed semantic proximities σ_{prec} and σ_{rec} are defined by:*

$$\sigma_{prec}(x\mathcal{R}y, x'\mathcal{R}'y') = \frac{|\mathcal{R} \cap \left(\mathcal{R}_1 \cdot \mathcal{R}' \cdot \mathcal{R}_2^{-1}\right)|}{|\mathcal{R}_1 \cdot \mathcal{R}' \cdot \mathcal{R}_2^{-1}|} \tag{9}$$

$$\sigma_{rec}(x\mathcal{R}y, x'\mathcal{R}'y') = \frac{|(\mathcal{R}_1^{-1} \cdot \mathcal{R} \cdot \mathcal{R}_2) \cap \mathcal{R}'|}{|\mathcal{R}_1^{-1} \cdot \mathcal{R} \cdot \mathcal{R}_2|} \tag{10}$$

The relaxed semantic proximity measures satisfy the local max-correctness and local max-completeness properties. As a consequence, they allow to provide relaxed semantic precision and recall measures which partially deals with alignment semantics. However, such measures do not consider the whole alignment semantic and then, they do not necessarily satisfy any property mentioned Section 3.1.

In our opinion, these semantic proximity measures are a first step for providing new semantic evaluations measures satisfying the desired properties. However, for satisfying these properties, it would be essential to propose new generalized precision and recall measures.

5.3 Restriction of ideal precision and recall

In order to deals with the semantic of alignments on one hand, and ideal precision and recall on the other hand, we first propose to use a partial closure of alignment instead of its full closure (α-consequence set). This partial closure has the advantage to be finite but in counterpart, it is defined relatively to a set of alignments. As a consequence, the ideal precision and recall can be computed, but their values depend on the set of considered alignments Λ. In a the case of evaluation campaigns, the set $\Lambda = A_{e_1} \cup ... \cup A_{e_n} \cup A_r$ will contain all correspondences provided by the participants, and the reference alignment.

Definition 12 (Bounded closure of an alignment). *The bounded closure of an alignment V given an alignment Λ ($V \subseteq \Lambda$) is defined as a set of correspondances issued from Λ which can be deduced from V.*

$$V^{+/\Lambda} = Cn(V) \cap \Lambda \tag{11}$$

The bounded closure $V^{+/\Lambda}$ of an alignment V is finite when Λ is finite (i.e. each alignment in Λ is finite). From this bounded closure definition, we provide Λ-bounded precision and recall.

Definition 13 (Λ-bounded precision measure). *Given a set of considered correspondences Λ, the precision of an alignment $A_e \subseteq \Lambda$ in comparison to a reference alignment $A_r \subseteq \Lambda$ is:*

$$P^{\Lambda}(A_e, A_r) = \frac{|A_e^{+/\Lambda} \cap A_r^{+/\Lambda}|}{|A_e^{+/\Lambda}|} \tag{12}$$

Definition 14 (Λ-bounded recall measure). *Given a set of considered correspondences Λ, the recall of an alignment $A_e \subseteq \Lambda$ in comparison to a reference alignment $A_r \subseteq \Lambda$ is:*

$$R^{\Lambda}(A_e, A_r) = \frac{|A_e^{+/\Lambda} \cap A_r^{+/\Lambda}|}{|A_r^{+/\Lambda}|} \tag{13}$$

With these measures the max-correctness, max-completeness, definiteness are verified. The semantic-identity property is also satisfied for each semantically equivalent alignments belonging to Λ (but not necessarily for the others). Still the overlapping-positiveness is not satisfied: $A_e^{+/\Lambda} \cap A_r^{+/\Lambda} = \emptyset \not\Longrightarrow Cn(A_e) \cap Cn(A_r) = \emptyset$

These measures are defined in the case of expressive alignments but they are dependent of Λ and consequently the precision and recall value are not absolute. Hence, these measures are useful for comparing a finite set of systems, but do not provide an absolute measure of precision and recall with regard to a reference alignment.

6 Conclusion

In this paper, we presented and analyzed several ontology alignment evaluation propositions. Actually, no concrete evaluation measure respects the semantic-equality and the overlapping-positiveness properties that an ideal semantic model should satisfy. More precisely, the semantic precision and recall measures cannot respect the semantic-equality due to the facts they still depend on the syntactic representation of alignments. To overcome these limitations, we first introduced alignment normalization principles which partially resolve the problem of semantic-equality. Then, we also proposed two new sets of evaluation measures. The first set of measures is built upon the generalized precision and recall framework and allows to locally consider the semantics of alignments. These measures can be seen as semantic-relaxed precision and recall. The second set of measures is proposed from an adaptation of ideal semantic measures. This adaptation makes the ideal semantic measures useable but in counterpart they do not verify the overlapping-positiveness property any more.

References

Jérôme David. *AROMA : une méthode pour la découverte d'alignements orientés entre ontologies à partir de règles d'association*. PhD thesis, Université de Nantes, 2007.

Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In Benjamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors, *Proceedings of the Workshop on Integrating Ontologies*, volume 156, pages 25–32. CEUR-WS.org, 2005.

Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI 07)*, pages 248–253, Hyderabad (IN), 2007.

Jérôme Euzenat. Algebras of ontology alignment relations. In *Proceedings of the 7th International Semantic Web Conference (ISWC 08)*, pages 387–402. Springer, 2008.

Alfred Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6(3):73–89, 1941.

Cornelis Joost (Keith) van Rijsbergen. *Information Retrieval*. Butterworths, London, 2 edition, 1979.

Towards a Benchmark for Instance Matching [⋆]

Alfio Ferrara, Davide Lorusso, Stefano Montanelli, Gaia Varese

Università degli Studi di Milano,
DICo, 10235 Milano, Italy,
{ferrara, lorusso, montanelli, varese}@dico.unimi.it

Abstract. In the general field of knowledge interoperability and ontology matching, instance matching is a crucial task for several applications, from identity recognition to data integration. The aim of instance matching is to detect instances referred to the same real-world object despite the differences among their descriptions. Algorithms and techniques for instance matching have been proposed in literature, however the problem of their evaluation is still open. Furthermore, a widely recognized problem in the Semantic Web in general is the lack of evaluation data. While OAEI (Ontology Alignment Evaluation Initiative) has provided a reference benchmark for concept matching, evaluation data for instance matching are still few. In this paper, we provide a benchmark for instance matching, with the goal of taking into account the main requirements that instance matching algorithms should address.

1 Introduction

The increasing popularity of Semantic Web technologies makes the ontology matching process a crucial task. Ontology matching [1] aim is to (semi) automatically detect semantic correspondences between heterogeneous ontologies. It can be performed at two different levels: schema matching and instance matching. The objective of *schema matching* [2] is to find out a set of mappings between concepts and properties in different ontologies, while the aim of *instance matching* is to detect instances referred to the same real-world object. When comparing different knowledge representations, ontologies' schemas should be merged, in terms of concepts and properties describing the domain. Then, mappings between different descriptions (i.e., ontologies' instances) of the same object should be discovered, in order to achieve the goal of providing a data integration system over Semantic Web sources.

Instance matching is also crucial in projects like OKKAM[1] [3], where the main idea is that real-world objects' descriptions could be retrieved, univocally identified and shared over the Web.

Most research has been focused on schema level matching, while instance matching problem has been mainly studied in the database field, in which it is more

[⋆] This paper has been partially funded by the BOEMIE Project, FP6-027538, 6th EU Framework Programme.

[1] http://www.okkam.org/.

specifically called *record linkage* problem [4–6]. However, as shown in the paper, instance matching brings new problems in comparison to record linkage and requires specific technologies.

2 The Instance Matching Problem

The instance matching problem is defined as follows. Given two instances $i1$ and $i2$, belonging to the same ontology or to different ontologies, instance matching is defined as a function $Im(i1, i2) \rightarrow \{0; 1\}$, where 1 denotes the fact that $i1$ and $i2$ are referred to the same real-world object and 0 denotes the fact that $i1$ and $i2$ are referred to different objects.
In order to find out properly if two individuals are referred to the same real-world object, an instance matching algorithm should satisfy different kinds of requirements. As shown in Figure 1, those can be divided in three main categories.

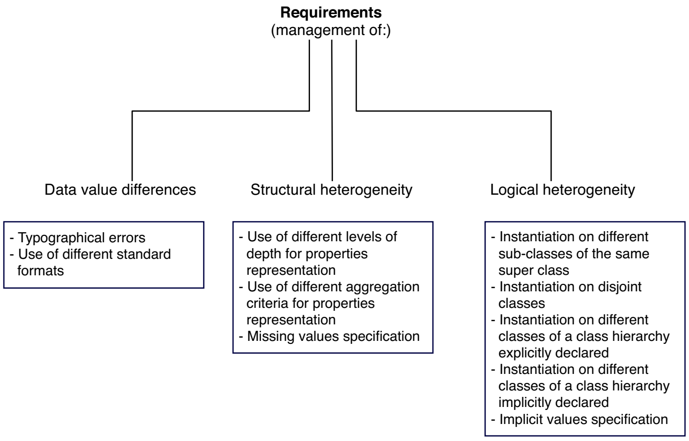

Requirements
(management of:)
Data value differences
Structural heterogeneity
Logical heterogeneity
- Typographical errors
- Use of different standard formats
- Use of different levels of depth for properties representation
- Use of different aggregation criteria for properties representation
- Missing values specification
- Instantiation on different sub-classes of the same super class
- Instantiation on disjoint classes
- Instantiation on different classes of a class hierarchy explicitly declared
- Instantiation on different classes of a class hierarchy implicitly declared
- Implicit values specification

Fig. 1. Instance matching requirements

Data value differences. An instance matching algorithm is required to recognize, as better as possible, corresponding values, even if data contain errors or are represented using different standard formats. This issue has been addressed in the field of record linkage research, and the problem of comparing instances' property values is the same as comparing records' attribute values.

Structural heterogeneity. Instances belonging to different ontologies can not only differ within their properties values, but they can also have different structures. While in record linkage the structure of records is usually given and schema and record matching are different problems, in instance matching, schema and instances are more strictly related. Thus, besides the capability to evaluate the level of similarity between property values, instance matching techniques have to go beyond heterogeneous individual representations by identifying the pairs of matching properties between two considered instances.

Logical heterogeneity. A specific ontologies' matching problem, which is not taken into consideration in record linkage process, is the need to infer implicit knowledge, typically referred to concepts hierarchy within the ontologies.

3 Design of a Benchmark for Instance Matching

A widely recognized problem in the Semantic Web is the lack of evaluation data. While OAEI (Ontology Alignment Evaluation Initiative)[2] [7] has provided a reference benchmark for concept matching, evaluation data for instance matching are still few. Further works dealing with concept matching evaluation are those published in ESWC 2008 [8, 9]. In particular, they argue that ontology matching techniques cannot be evaluated in an application independent way, since the same matching technique can produce different quality results based on the end-to-end application that exploits the alignments.
In this paper, we provide a benchmark for instance matching. The aim of our benchmark is to take into account all the main requirements presented in the previous section and to provide a complete set of tests for instance matching algorithms evaluation. A contribution of our work is not only the definition of a specific benchmark, but also the definition of a semi-automatic procedure for the generation of several different benchmarks. In Figure 2, the overall process of benchmarks generation is shown. As an example of this general procedure, we describe in the following a specific instantiation of it, that is the creation of a specific benchmark for instance matching. That benchmark is available at *http://islab.dico.unimi.it/iimb/.*

3.1 Reference ABox Generation

First of all, we chose a domain of interest (i.e., the domain of movie data), and we created a reference ($\mathcal{ALCF}(D)$) TBox for it, based on our knowledge of the domain. The reference TBox is available at *http://islab.dico.unimi.it/ontologies/-benchmark/imdbT.owl.* This contains 15 named classes, 5 object properties and 13 datatype properties. The reference TBox is then populated by automatically creating a reference ABox. Data are extracted from IMDb [3] by executing a query

[2] http://oaei.ontologymatching.org/2007/benchmarks/.

[3] http://www.imdb.com/.

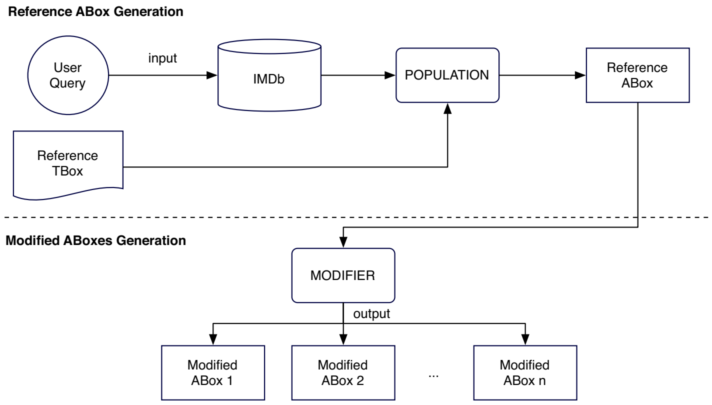


Fig. 2. Benchmarks generation

Q of the form:

$$SELECT * FROM\ movies\ WHERE\ title\ LIKE\ '\%X\%'$$

where X is a variable specifying a word of our choice. Thus, all selected movies contain the word X in their title. The corresponding individuals in the reference ABox are referred to similar objects, but each of them represents a distinct object in the real world. As a consequence, each instance can be univocally identified. In order to get our reference ABox, we put $X = Scarface$. The reference ABox obtained in that way contains 302 individuals, that is all the movie objects matching the query and all the actors in the movie cast.

3.2 Modified ABoxes Generation

Once the reference ABox is created, we generate a set of modified ABoxes, each consisting in a collection of instances obtained modifying the corresponding instances in the reference ABox. Transformations introduced in benchmark ABoxes can be distinguished into three main categories. In particular, each modification category simulates a specific problem that can be found when comparing ontologies' instances, that is the issues discussed in section 2. Modifications belonging to different categories are also combined together within the same ABox.

4 Generating Instance Modifications

In this section, we describe the *Modifier* module of our benchmarks generation procedure, that is the way the modified ABoxes of benchmarks are generated. Given the reference ABox as input, and a user specification of all the transformations to apply on it, the *Modifier* module automatically produces the corresponding modified ABoxes. In the following, all the modifications that can be applied on the reference ABox are presented.

4.1 Data Value Differences

The goal of this first category of modifications is to simulate the differences that can be found between instances referred to the same object at the property value level. Those include typographical errors, use of different standard formats to represent the same value, or a combination of both within the same value.

Typographical errors. Real data are often dirty. That is mainly due to typographical errors made by humans while storing data.
In order to simulate typographical errors, we use a function that takes as input a datatype property value and produces as output a modified value. This kind of transformation can be applied to each datatype property value (e.g., string value, integer value, date value). The modifications to apply on the input value are randomly chosen between the following:

- *Insert character.* A random character (or a random number, if the property has a numerical value) is inserted in the input value at a random position.
- *Modify character.* A random character (or a random number, if the property has a numerical value) is modified in the input value.
- *Delete character.* A random character (or a random number, if the property has a numerical value) is deleted in the input value.
- *Exchange characters' position.* The position of two adjacent characters (or two adjacent numbers, if the property has a numerical value) is exchanged in the input value.

For example, the movie title "Scarface" can be transformed in the modified value "Scrface", obtained deleting a random character from the original string.
In addition, it is possible to specify the level of severity (i.e., low, medium or high) in applying such transformations. Anyway, the number of transformations introduced in the input value is proportional to the value's length. If the number of transformations to apply is greater than one, the corresponding value can be modified combining different transformations.
Typographical modifications can be applied to "identifying properties", "non-identifying properties" or both. That classification is based on the analysis of the percentage of null and distinct values specified for the selected property. In particular, properties with an high percentage of distinct values and a low percentage of null values are classified as the most identifying.
Of course, the total amount of modifications applied to each modified ABox has to change the reference ABox in a way that it is still reasonable to consider the two ABoxes semantically equivalent. In other words, a modified ABox is included in the benchmark only if a human can understand that its instances are referred to the same real-world objects as the ones belonging to the reference ABox. Thus, in order to evaluate the distance between the reference ABox and each modified ABox, we introduce a measure that takes into account the number of modifications applied to the same ABox, the kind of the properties (i.e., "identifying properties" or "non-identifying properties") which have been

modified, and the level of severity of the modifications (i.e., low, medium or high). However, this measure does not affect the instance matching results in a deterministic way, since they depend on the weight that the tested algorithm gives to each kind of modification. Anyway, we assume that a modified ABox can be considered semantically equivalent to the reference ABox only if it changes no more than 20% of each instance description.

Use of different standard formats. The same data within different sources can be represented in different ways.
In order to simulate the use of different standards within different sources, we use a function that takes as input a property value which allows standard modifications (e.g., person name) and produces as output a modified value, using a different standard format. For example, the director name "De Palma, Brian" can be transformed in the modified value "Brian De Palma", which is another standard format to specify a person name.

4.2 Structural Heterogeneity

Another kind of situation that is simulated in our instance matching benchmark is the comparison between instances with different schemas. In fact, even assuming that concept mappings are available, the same individual feature (i.e., each instance property) can be modeled in different ways. Moreover, different descriptions of the same real-world object can specify different subsets, eventually empty, of all the possible values for that property. Combinations of different transformations belonging to this class of modification are also applied in the benchmark.

Use of different levels of depth for properties representation. A first example of this class of heterogeneity is shown in Figure 3. The two instances

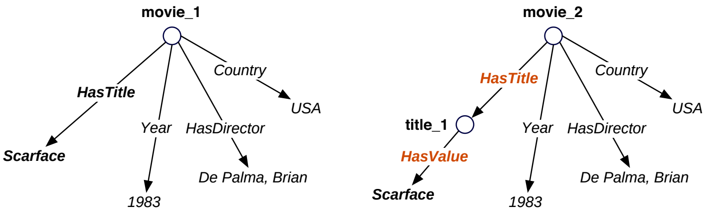


Fig. 3. Use of different levels of depth to represent the same property

*movie*_1 and *movie*_2 are both referred to the same film, but the movie title property is modeled in two different ways. In fact, the title of *movie*_1 is specified directly through a datatype property value, while the title of *movie*_2 is

specified through a reference to another individual which has a property with the same title value (i.e., "Scarface"). In particular, in the first representation, the property *HasTitle* is a datatype property, while in the second one it is an object property and its value is the reference to *title_1* instance.
In order to simulate the comparison between instances with different schemas, we use a function that takes as input a datatype property and produces as output an object property with the same name. Moreover, the function creates a new attribute to the generated object property, whose value is the same as the original datatype property.

Use of different aggregation criteria for properties representation. In an analogous way, the name of a person can be stored all within the same property, or it can be split into different properties such as, for example, *Name* and *Surname*. Figure 4 shows two different ways of modeling the name "Pacino, Al". In the first representation the whole value is stored within the property *Name*, while in the second one the string is split into the two values "Pacino" and "Al", referred to the properties *Name* and *Surname* respectively.

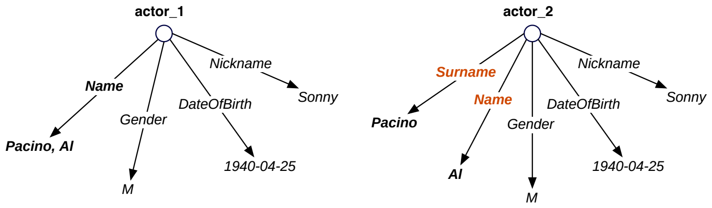


Fig. 4. Use of different aggregation criteria to represent the same property

In order to simulate the comparison between properties modeled in different ways, we use a function that takes as input a datatype property value that can be split and produces as output two new datatype properties, each specifying a different part of the original value.

Missing values specification. A further example of structural heterogeneity is shown in Figure 5. The two instances *movie_1* and *movie_2* are both referred to the same film, but the two different descriptions specify different subsets of values on the property *Genre*.
In order to simulate the comparison between different sets of values referred to the same property, we use a function that takes as input the set of values specified for a selected property and produces as output a subset, eventually empty, of it. This kind of transformation can be applied to each property. Moreover, if a property allows multiple values, it is possible to specify if deleting all the values of the selected property or a random number of them.

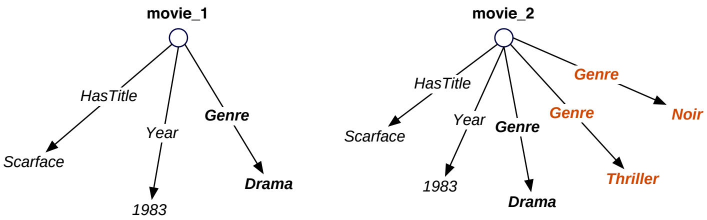


Fig. 5. Specification of different subsets of values on the same multi-values property

4.3 Logical Heterogeneity

Finally, instance matching process should take into account the need to use some kind of reasoning, in order to find out correctly instances to be compared. In fact, ontologies' individuals referring to the same entity can be instantiated in different ways within different ontologies. In the following we describe five kinds of situations that we develop in our benchmark, that can also be combined together. Each requires some kind of reasoning. Examples of those are shown in Figure 6.

Reference TBox

$Movie \sqsubseteq Item$	$Movie \sqcap Product \sqsubseteq \bot$
$Film \sqsubseteq Item$	$Movie \equiv \forall p.G$
$Product \sqsubseteq Item$	$SubM \equiv \forall p.SubG$
$Action \sqsubseteq Movie$	$SubG \sqsubseteq G$

Reference ABox	**Modified ABox**
$movie\_1 : Movie$	$movie\_1 : Film$
$movie\_2 : Movie$	$movie\_2 : Product$
$movie\_3 : Movie$	$movie\_3 : Action$
$movie\_4 : Movie$	$movie\_4 : SubM$
$movie\_5 : Movie$	$movie\_5 : Movie$
$(movie\_5, \text{“}Scarface''): HasTitle$	$movie\_5 : (\exists HasTitle.\text{“}Scarface'')$

Fig. 6. Examples of logical heterogeneity

Instantiation on different subclasses of the same superclass. This transformation is obtained instantiating identical individuals into different subclasses of the same class. For example, in our benchmark, all the movie objects are instances of class *Movie* in the reference ABox. Instead, in one of the modified ABoxes, we change the type of those individuals, making them instances of class *Film*. Classes *Movie* and *Film* are both subclasses of *Item*. In Figure 6, $movie\_1$

is instance of *Movie* in the reference ABox, while it is instance of *Film* in the modified ABox. Instance matching algorithms are thus required to recognize that those two instances are referred to the same object, even if they belong to different concepts.

Instantiation on disjoint classes. This transformation is obtained instantiating identical individuals into disjoint classes. For example, in one of the modified ABoxes, we change the type of all the movie objects, making them instances of class *Product*. Classes *Movie* and *Product* are defined as disjoint classes in the reference TBox. In Figure 6, *movie_2* is instance of *Movie* in the reference ABox, while it is instance of *Product* in the modified ABox. In this case we want that tested algorithms would be able to recognize that instances belonging to disjoint classes cannot be referred to the same real-world object, even if they seem identical.

Instantiation on different classes of a class hierarchy explicitly declared. This transformation is obtained instantiating identical individuals into different classes on which an explicit class hierarchy is defined. For example, an individual representing a movie can be classified as an instance of the general concept *Movie*, as it is in the reference ABox, or it can be classified as an instance of a more specific subclass of it, such as *Action*, *Biography*, *Comedy* or *Drama*, depending on the value that the movie instances specify on the property *Genre*. In Figure 6, *movie_3* is instance of *Movie* in the reference ABox, while it is instance of its subclass *Action* in the modified ABox, since it is an action movie. Instance matching algorithms are thus required to recognize that those two instances are referred to the same object, even if they belong to different concepts within the class hierarchy. This explicit class hierarchy declaration can be recognized using a RDFS reasoner.

Instantiation on different classes of a class hierarchy implicitly declared. A further modification that we apply in the benchmark is the instantiation of identical individuals into different classes on which an implicit class hierarchy is defined. Such an implicit class hierarchy declaration can be obtained through the use of restrictions. For example, the restrictions specified on classes *Movie* and *SubM* in the reference TBox, implicitly declare that *SubM* is a subclass of *Movie*. In Figure 6, *movie_4* is instance of *Movie* in the reference ABox, while it is instance of *SubM* in the modified ABox. Instance matching algorithms are thus required to recognize that those two instances are referred to the same object, even if they belong to different concepts which are not explicitly related. This implicit class hierarchy declaration can be recognized using a DL reasoner.

Implicit values specification. Another use of restrictions that requires a reasoning process, is the comparison between an explicit specified value and an implicit specified one, that is using an *hasValue* restriction. This kind of situation is simulated in our benchmark by adding a new type for each instance of the modified ABox. This type is a class that (implicitly) specifies property values through an *hasValue* restriction. In Figure 6, in the reference ABox, *movie_5* is instance of *Movie* and its value on the property *HasTitle* is “Scarface”; in the modified ABox, *movie_5* is as well instance of *Movie*, but it is also instance of the restriction class that implicitly specifies the value “Scarface” for its *HasTitle* property. Instance matching algorithms are thus required to recognize that those two instances are referred to the same object, even if some property values of the modified instance are implicitly defined.

5 Benchmark at Work

In this section, we describe how the generated benchmark is used to evaluate instance matching algorithms. Each execution of the evaluation process takes as input a couple of ABoxes, that is the reference ABox and one of the modified ABoxes, and produces the set of instance mappings found by the tested algorithm. The output mapping alignment is then compared with the expected one, which is given together with each modified ABox. That reference alignment is automatically generated by specifying a mapping for each couple of corresponding instances, that is the one belonging to the reference ABox and the one obtained by applying to it one or more of the modifications discussed in section 4.
Instance matching algorithms are evaluated according to the following parameters.

- *Precision:* the number of correct retrieved mappings / the number of retrieved mappings.
- *Recall:* the number of correct retrieved mappings / the number of expected mappings.
- *F-measure:* 2 · (precision · recall) / (precision + recall).
- *Fall-out:* the number of incorrect retrieved mappings / the number of non-expected mappings.
- *Execution time:* time taken by the tested algorithm to compare the two input ABoxes. This parameter measures how well the tested algorithm scales.

As an example, the results obtained by two instance matching algorithms are reported. Figure 7 shows the precision and recall evaluation of the two instance matching algorithms over the generated benchmark, distinguishing the results obtained in the three main classes of problems simulated in our benchmark (i.e., data value differences, structural heterogeneity, logical heterogeneity) and the ones obtained executing each algorithm without using any reasoner and using a (DL) reasoner (i.e., Pellet). The results obtained comparing the reference ABox with modified ABoxes simulating data value differences are higher than the ones obtained in the other categories, since string matching techniques are quite consolidated. The results obtained comparing the reference ABox with modified ABoxes simulating structural heterogeneity are not very high because neither the first nor the second algorithm can manage the use of different aggregation criteria

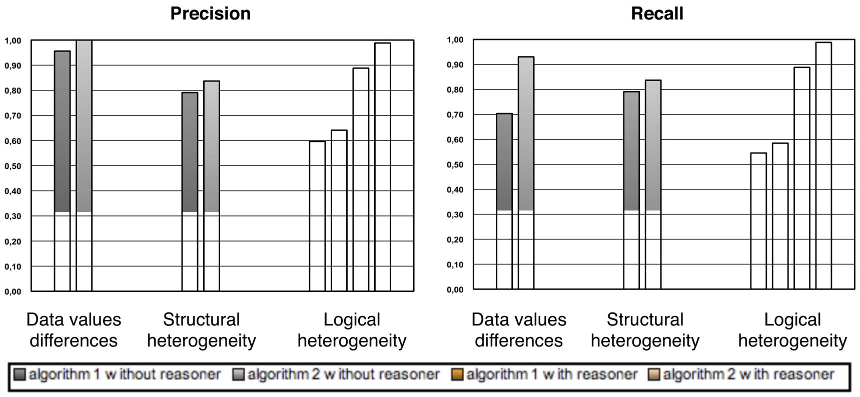


Fig. 7. Precision and recall evaluation

for properties representation. The results obtained comparing the reference ABox with modified ABoxes simulating logical heterogeneity are greatly affected by the use of a reasoner.

Finally, in Figure 8, the overall results obtained executing the two algorithms (with reasoner) on our benchmark are reported. That test had been executed on a Pentium 4 (2.00 GHz) with 512 MB of RAM. For each pair of compared

IM Algorithm	Precision	Recall	F-measure	Fall-out	Execution time
algorithm_1	0.88	0.79	0.81	0.05	50 sec
algorithm_2	0.94	0.92	0.93	0.01	31 sec

Fig. 8. Overall evaluation of two instance matching algorithms

instances, the first algorithm [10] analyzes all their property values, while the second algorithm [11] checks only the values specified for the "most identifying" properties. That is why the execution time of the first algorithm is greater than the execution time of the second one. Moreover, the recall of the second algorithm is higher than the recall of the first one due to the fact that all the modifications applied to "non-identifying" properties are ignored. A more detailed description of the two algorithms is available in [10, 11].

6 Concluding Remarks

In this paper, we provided a benchmark for instance matching, taking into account the main requirements that instance matching algorithms should address.

A contribution of our work is not only the definition of a specific benchmark, but also the definition of a semi-automatic procedure for the generation of several different benchmarks.

Future works include the creation of further benchmarks dealing with data belonging to different sources and different domains. In particular, we would like to create a benchmark in which data belonging to different sources but referred to the same real-world objects are compared. For example, it can include a mapping between movie descriptions in IMDb and Amazon. In that case, the expected alignments have to be done manually, so the benchmark dimension cannot be significant for a real benchmark. However, it would be interesting to compare the results obtained by the same algorithms executing that benchmark and our semi-automatically generated one, in order to evaluate the quality of our benchmark generation itself.

Another possible development would be the definition of a set of rules that automatically choose the modifications to apply, for each modified ABox, to the reference ABox.

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag (2007)
2. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics (JoDS) (2005)
3. Bouquet, P., Stoermer, H., Niederee, C., Mana, A.: Entity name system: The backbone of an open and scalable web of data. In: Proceedings of the IEEE International Conference on Semantic Computing, ICSC 2008, IEEE (2008)
4. Fellegi, I., Sunter, A.: A theory for record linkage. J. Am. Statistical Assoc. (1969)
5. Winkler, W.: The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, Wachington, DC (1999)
6. Gu, L., Baxter, R., Vickers, D., Rainsford, C.: Record linkage: Current practice and future directions. Technical report, CSIRO Mathematical and Information Sciences, Canberra, Australia (2003)
7. Shvaiko, P., Euzenat, J., Noy, N., Stuckenschmidt, H., Benjamins, V., Uschold, M.: Proceedings of the 1st international workshop on ontology matching (om-2006) collocated with the 5th international semantic web conference (iswc-2006), athens, georgia, usa, november 5, 2006. In: OM. Volume 225., CEUR-WS.org (2006)
8. Hollink, L., van Assem, M., Wang, S., Isaac, A., Schreiber, G.: Two variations on ontology alignment evaluation: Methodological issues. ESWC 2008 (2008)
9. Isaac, A., Matthezing, H., van der Meij, L., Schlobach, S., Wang, S., Zinn, C.: Putting ontology alignment in contex: Usage scenarios, deployment and evaluation in a library case. ESWC 2008 (2008)
10. Castano, S., Ferrara, A., Montanelli, S.: Matching ontologies in open networked systems: Techniques and applications. Journal on Data Semantics (JoDS) (2006)
11. Bruno, S., Castano, S., Ferrara, A., Lorusso, D., Messa, G., Montanelli, S.: Ontology coordination tools: Version 2. Technical Report D4.7, BOEMIE Project, FP6-027538, 6th EU Framework Programme (2007)

Using quantitative aspects of alignment generation for argumentation on mappings

Antoine Isaac[1], Cássia Trojahn[2], Shenghui Wang[1], Paulo Quaresma[2]

[1] Vrije Universiteit, Department of Computer Science, Amsterdam, Netherlands
[2] University of Évora, Department of Informatics, Évora, Portugal

Abstract. State-of-the art mappers articulate several techniques using different sources of knowledge in an unified process. An important issue of ontology mapping is to find ways of choosing among many techniques and their variations, and then combining their results. For this, an innovative and promising option is to use frameworks dealing with arguments for or against correspondences. In this paper, we re-use an argumentation framework that considers the confidence levels of mapping arguments. We also propose new frameworks that use voting as a way to cope with various degrees of consensus among arguments. We compare these frameworks by evaluating their application to a range of individual mappers, in the context of a real-world library case.

1 Introduction

An important problem for ontology alignment is to find ways of choosing among the many tools and techniques available and their variations, and then combining their results. This is almost infeasible by purely manual efforts, and fixed heuristics for combining a pre-selected set of mappers will not fit a situation where more and more matching tools and options can be applied to an even greater variety of cases.

A first range of methods relies on (partial) evaluation of the results given by different techniques so as to *recommend* the best performing ones for the case at hand [1, 2]. Others anticipate such results by comparing the characteristics of the considered alignment case with "profiles" of matchers, as determined by previous evaluation [3]. However, these methods result in applying the same treatment to all the mappings obtained by a same method; they do not allow for considering each mapping. In the context of peer-to-peer systems, a more flexible approach has been proposed [4] that explores the way peers agree on a set of mappings, by evaluating the translations resulted from the application of each mapping when one peer queries for information provided by another.

A promising option is to use argumentation frameworks where arguments in favour or against mappings between concepts are declaratively represented and processed [5, 6]. Here, a set of mappers, representing different alignment approaches, generate a set of arguments that support the mappings. According to the definition of attacking relations, an argument for a mapping generated by one mapper can be supported or attacked by other arguments from other mappers. Based on the framework instantiation (using specific attacking relation and preference order), it is possible to compute globally acceptable mappings.

These argumentation frameworks consider however the arguments based on their *intention* only. An argument against a concept mapping can successfully attack all the arguments in favour of it, even if there are dozens of these. In this paper, we investigate *quantitative* aspects of alignment generation among a set of arguing mappers. We focus especially on investigating and comparing the value, for the argumentation process, of alignment generation: (1) confidence level: can we use the confidence level of the mappings to solve argumentation conflicts? ; (2) consensus among mappers: can we use the agreement between mappers to measure the validity of the mappings in question?

In this paper, we re-use an argumentation framework that considers the confidence levels of mapping arguments [5]. We also propose new frameworks that use voting as a way to cope with various degrees of support for arguments. We compare these frameworks by evaluating their application to a range of state-of-the-art individual mappers, in the context of a real-world library case.

2 Argumentation Frameworks

The framework we have re-used and extended to deal with consensus, S-VAF, is based on Value-based Argumentation, itself based on Dung's classical system. In this section we present these three frameworks, as well as our new proposals.

2.1 Classical argumentation framework

Dung, observing that the core notion of argumentation lies in the opposition between arguments and counter-arguments, defines an argumentation framework (AF) as follows:

Def. [7] An Argumentation Framework is a pair $AF = (AR, attacks)$, AR is a set of arguments and $attacks$ is a binary relation on AR.

$attacks(a, b)$ means that the argument a attacks the argument b. A set of arguments S attacks an argument b if b is attacked by an argument in S. The key question about the framework is whether a given argument $a \in AR$ should be accepted or not. Dung proposes that an argument should be accepted only if every attack on it is rebutted by an accepted argument. This notion then leads to the definition of acceptability (for an argument), admissibility (for a set of arguments) and preferred extension:

Def. [7] An argument $a \in AR$ is *acceptable* with respect to set arguments S, noted $acceptable(a, S)$, if $\forall x \in AR\ (attacks(x, a) \longrightarrow \exists y \in S, attacks(y, x))$

Def. [7] A set S of arguments is *conflict-free* if $\neg\ \exists x, y \in S$, $attacks(x, y)$. A conflict-free set of arguments S is *admissible* if $\forall x \in S$, $acceptable(x, S)$. A set of arguments S is a *preferred extension* if it is a maximal (with respect to set inclusion) admissible set of AR.

A preferred extension represents a consistent position within AF, which defends itself against all attacks and cannot be extended without raising conflicts.

2.2 Value-based argumentation framework

In Dung's framework, all arguments have equal strength, and attacks always succeed, except if the attacking argument is otherwise defeated. However, as noted in [8], in many domains, including ontology alignment, arguments may provide reasons which may be more or less persuasive. Moreover, their persuasiveness may vary according to their audience. Bench-Capon has extended the notion of AF so as to associate arguments with the social values they advance:

Def. [9] A Value-based Argumentation Framework (VAF) is a 5-tuple $VAF = (AR, attacks, V, val, P)$ where $(AR, attacks)$ is an argumentation framework, V is a nonempty set of values, val is a function which maps elements of AR to elements of V and P is a set of possible audiences.

Practically, in [6], the role of value is played by the types of ontology match that ground the arguments, covering general categories of matching approaches: semantic, structural, terminological and extensional. We argue further — and will use later — that any kind of matching ground identified during a mapping process or any specific matching tools may give rise to a value. The only limitations are (i) a value can be identified and shared by a source of mapping arguments and the audience considering this information (ii) audiences can give preferences to the values. An extension to this framework, required for deploying argumentation processes, indeed allows to represent how audiences with different interests can grant preferences to specific values:

Def. [9] An Audience-specific Value-based Argumentation Framework (AVAF) is a 5-tuple $VAF_p = (AR, attacks, V, val, valpref_{aud})$ where AR, $attacks$, V and val are as for a VAF, aud is an audience and $valpref_{aud}$ is a preference relation (transitive, irreflexive and asymmetric), $valpref_{aud} \subseteq V \times V$.

$valpref_{aud}(v_1, v_2)$ means that audience aud prefers v_1 over v_2. Attacks are then deemed successful based on the preference ordering on the arguments' values. This leads to re-defining the notions seen previously:

Def. [9] An argument $a \in AR$ defeats an argument $b \in AR$ for audience aud, noted $defeats_{aud}(a, b)$, if and only if both $attacks(a, b)$ and not $valpref_{aud}(val(b), val(a))$. An argument $a \in AR$ is *acceptable* to audience aud with respect to a set of arguments S, noted $acceptable_{aud}(a, S)$, if $\forall x \in AR,\ defeats_{aud}(x, a) \longrightarrow \exists y \in S,\ defeats_{aud}(y, x)$.

Def. [9] A set S of arguments is *conflict-free* for audience aud if $\forall x, y \in S,\ \neg attacks(x, y) \vee valpref_{aud}(val(y), val(x))$. A *conflict-free* set of arguments S for aud is *admissible* for aud if $\forall x \in S,\ acceptable_{aud}(x, S)$. A set of arguments S in the VAF is a *preferred extension* for audience aud if it is a maximal admissible set (with respect to set inclusion) for aud.

In order to determine preferred extensions with respect to a value ordering promoted by distinct audiences, *objective* and *subjective* acceptance are defined:

Def. [9, 6] An argument $a \in AR$ is *subjectively acceptable* if and only if a appears in the preferred extension for some specific audiences. An argument $a \in AR$ is *objectively acceptable* if and only if a appears in the preferred extension for every specific audience.

2.3 Strength-based Argumentation Framework

Value-based argumentation acknowledges the importance of preferences when considering arguments. However, in the specific context of ontology alignment, an objection can still be raised about the lack of complete mechanisms for handling persuasiveness. Indeed, off-the-shelf matching tools very often provide a mapping with a measure that reflects the strength of the similarity between the two entities, or a more general confidence they have in the mapping – almost always it is provided without any detail allowing to distinguish between the two. These measures – we will use *strength* in the following – are usually derived from similarity assessments made during the alignment process, *e.g.* from edit distance measure between labels, or overlap measure between instance sets, as in [10]. They are therefore often based on objective grounds.

However, there is no objective theory nor even informal guidelines for determining such strengths. Using them to compare results from different mappers is therefore questionable especially because of potential scale mismatches. For example, a same strength of 0.8 may not correspond to the same level of confidence for two different mapper.

It is one of our goals to investigate whether considering strengths gives better results or not.[3] To this end, we adapt a formulation introduced in [11, 5] to consider the strength granted to mappings for determining attacks' success:

Def. A Strength and value-based Argumentation Framework (S-VAF) is a 6-tuple $(AR, attacks, V, val, P, str)$ where $(AR, attacks, V, val, P)$ is a value-based argumentation framework, and str is a function which maps elements of AR to real values from the interval $[0, 1]$, representing the *strength* of the argument. An audience-specific S-VAF is an S-VAF where the generic set of audiences is replaced by the definition of a specific $valpref_{aud}$ preference relation over V.

Def. In an audience-specific S-VAF, an argument $a \in AR$ defeats an argument $b \in AR$ for audience aud if and only if $attacks(a, b) \wedge (\ str(a) > str(b) \vee (str(a) = str(b) \wedge valpref_{aud}(val(a), val(b)))\ )$

In other words, for a given audience, an attack succeeds if the strength of the attacking argument is greater than the strength of the attacked one; or, if both arguments have equal strength, the attacked argument is not preferred over the attacking argument by the concerned audience. Similarly to what is done for VAFs, an argument is acceptable for a given audience *w.r.t* a set of arguments if every argument defeating it is defeated by other members of the set. A set of arguments is conflict-free if no two members can defeat each other. Such a set is admissible for an audience if all its members are acceptable for this audience *w.r.t* itself. A set of arguments is a preferred extension for an audience if it is a maximal admissible set for this audience.

[3] Note that as opposed to what is done [11, 5] this paper aims at experimenting with mappers that were developed prior to the experiment, and hence more likely to present strength mismatches.

2.4 Argumentation Frameworks with voting

The previously described frameworks capture the possible conflicts between mappers, and find a way to solve them. However, they still fail at rendering the fact that sources of mappings often agree on their results, and that this agreement can be meaningful. Some large-scale experiments involving several alignment tools – as the OAEI 2006 Food track campaign [12] – have indeed shown that the more often a mapping is agreed on, the more chances for it to be valid.

In the following, we adapt the S-VAF presented above to consider the level of consensus between the sources of the mappings, by introducing *voting* into the definition of successful attacks. We first describe the notion of *support* which enables arguments to be counted as defenders or co-attackers during an attack:

Def. A *Support-aware Framework* (Sup-VAF) is a 7-tuple *(AR, attacks, supports, V,val,P,str)* where $(AR, attacks, V, val, P, str)$ is a S-VAF, and *supports* and *attacks* are disjoint (reflexive) binary relations over AR.

The voting is used to determine whether an attack is successful or not. Our first proposal opts for a simple voting scheme, where the number of supporters decides for success — as done in the *plurality voting system.*

Def. In a *Simple plurality voting Sup-VAF* an argument $a \in AR$ $defeats_{aud}$ an argument $b \in AR$ for audience aud if and only if $attacks(a, b) \quad \wedge \quad (\quad |\{x|supports(x, a)\}| > |\{y|supports(y, b)\}| \quad \vee \quad (|\{x|supports(x, a)\}| = |\{y|supports(y, b)\}| \wedge valpref_{aud}(val(a), val(b))) \quad)$.

This voting mechanism is based on simple counting. In fact, as we have seen previously, mappers sometimes return mappings together with a confidence value. There are voting mechanisms which address this confidence information. The first and most elementary one would be to sum up the strengths of supporting arguments. However, as for the S-VAF, this would rely on the assumption that the strengths assigned by different mappers are similarly scaled, which as we have seen is debatable in practice.

One possible option is to consider rankings derived from those confidence levels. First, we rank arguments on a value basis. For a given value $v \in V$, we define a function $rank_v : AR \longrightarrow \mathbb{N}$ that enables to order all the arguments according to their strength. Practically we choose to count, for each arguments, the ones that have a lower confidence level: $rank_v(a) = |\{x \in AR|val(x) = v \wedge str(x) < str(a)\}|$. Notice that this "ranking" reflects a partial order, as it allows for ties (for mappings with a same strength). It however avoids turning to random ordering decisions, and allows for seamless ranking of arguments derived from mappings that were not given any strength, by just considering that these arguments have an infinitely low strength. Based on this ranking, it is possible to define a voting process inspired by the Borda count method, which is one the reference methods for aggregating ranked choices – for each argument, we average the ranks given to it by the audiences which support it: [13]:

Def. In a *Borda count Sup-VAF* an argument $a \in AR$ $defeats_{aud}$ an argument $b \in AR$ for audience aud if and only if

$$attacks(a, b) \wedge (\ bordaCount(a) > bordaCount(b) \vee$$
$$(\ bordaCount(a) = bordaCount(b)\ \wedge\ valpref_{aud}(val(a), val(b))\ )\ ),$$
where
$$bordaCount(arg) = \frac{\sum_{\{x|supports(x,arg)\}} rank_{val(x)}(x)}{|\{x|supports(x, arg)\}|}.$$

3 Experiments

3.1 Experiment case

Our testbed reproduces the Library Track of the 2007 OAEI campaign.[4] The National Library of the Netherlands maintains two book collections, each annotated with one thesaurus – *GTT* (35K concepts) and *Brinkman* (5K). These thesauri have to be aligned with links that correspond to classical thesaurus relations (*broadMatch*, *narrowMatch*, *relatedMatch*) or to semantic equivalence (*exactMatch*). It is important to mention that among the 2.4 Million of books in the two collections, 250K are actually dually annotated by both thesauri.

3.2 Mappers used

To carry out our experiments, we have selected the results of six mappers, which we believe to be a realistic sample of the available technology. The first three are state-of-the-art mappers developed by the community (OAEI participants), while the others result from our previous work. They exhibit a balance between generic methods – *e.g.*, string edit distance – and strategies that are arguably more appropriate to the case at hand – *e.g.*, using Dutch lexical knowledge.

OAEI participants. The first group of mappers we used are the participants of the OAEI Library Track: **Falcon** [14], **DSSim** [15] and **Silas** [16]. These tools are *hybrid*, as they use several alignment techniques in an integrated process. For instance, Falcon considers the similarity of both lexical and structural information of concepts, while Silas combines lexical techniques with applying instance-based similarity measures on books descriptions accessed from a library service. Note that, as generic matchers, they mainly return equivalence (*exactMatch*) mappings, except Silas, which provides a significant number of *related* matches.

"Homegrown" mappers. We also re-used mappers developed for previous experiments. First, an **edit-distance lexical mapper** applies string similiarity to (tokenized) labels, resulting in various exact equivalent, broader, narrower and related weighted matches. Second, a **Dutch SKOS lexical mapper** outputs weighted equivalent and broader mappings, based on Dutch morphological knowledge, exploiting the different type of labels of concepts as represented in SKOS. Third, an **extensional mapper** exploits the simple co-occurrence of concepts in KB book annotations [10] to produce weighted equivalence links. For more details, see `http://www.few.vu.nl/~aisaac/om2008/mappers-om08.pdf`.

[4] `http://oaei.ontologymatching.org/2007/library`

3.3 Evaluation measures

We set our evaluation in a scenario where mappings are used to translate book annotations from one thesaurus to the other [17]. One mapping – it is of course possible to restrict the mappings by selecting only one kind of relation, for instance *exactMatch* – is considered as a translation rule, which translates one GTT concept into its corresponding Brinkman concept. All mappings which involve the same GTT concept are aggregated into a single rule.

To carry out our evaluation, we use the 250K dually annotated books we have mentioned as a golden standard. For one such book, if one of its GTT annotation concept has a translation rule, we consider this book can be *fired*. Each of its GTT annotation concepts is then translated into its Brinkman correspondence(s). The original Brinkman annotation is taken as a gold standard, which is used to measure the quality of the generated mappings.

We measure how many translated concepts are correct (precision), how many real Brinkman annotation concepts are missed (recall), and a Jaccard overlap as combined measure of these two:

$$P_a = \frac{\sum \frac{\#correct}{|B_t|}}{\#books\_fired}, \quad R_a = \frac{\sum \frac{\#correct}{|B_o|}}{\#all\_books}, \quad J_a = \frac{\sum \frac{\#correct}{|B_o \cup B_t|}}{\#all\_books}$$

where $\#correct$ is the number of translated Brinkman concepts actually used, B_o and B_t are the original and translated Brinkman annotation, respectively.

3.4 Argumentation settings

Characterisation of mapping arguments and attacking relation. All the mappers we used return correspondences in the form of $m = (e_1, e_2, s, r)$, where e_1 and e_2 are entities from the two ontologies, s a confidence level, and r a mapping relation — *exactMatch*, *broadMatch*, *narrowMatch* or *relatedMatch*. Following [6, 5], arguments were created from these correspondences, as 6-tuples $arg = (e_1, e_2, s, r, v, h)$ where v denotes a value or type of mapping argument (here, the tool which created the mapping) and h a support token (+ or $-$, depending on whether the argument supports the correspondence or not). An *attack* relationship holds between two arguments if these involve the same pair of concepts but exhibit opposite support tokens.

Generating negative arguments. Our problem is to define the arguments which are *against* a given correspondence. The results of most of the state-of-the-art tools must be interpreted as supporting correspondences; except in some formal approaches, there is no "negative mapping". [6] solves this by examining the features of the concepts, such as their label or position in the ontologies' structural network, and use OWL semantics to find whether agents argue for or against a correspondence. In practice, this complex process amounts to re-define a mapping step, as the strategy and material used are very similar to the ones exploited by the individual mappers. Here, we propose to experiment with two simpler strategies which do not require to investigate the alignment space again.

Negative arguments as failure (NAF). This basic strategy relies on the assumption that mappers return *complete* results. For every possible pair of concepts and mapping relation, we check whether a mapper outputs it. If not, this correspondence is considered to be at risk, and a negative argument is generated, with an arbitrary strength of 1. This assumption, at first sight quite bold, is nevertheless supported by the observation that most mappers try to provide as many mappings as possible, the amount of (equivalent) mapping pairs being comparable to the size of the smallest ontology aligned.

Negative arguments based on relation disjointness (NARD). The second strategy assumes that two different thesaurus-inspired mapping relations (*broadMatch*, *narrowMatch* or *relatedMatch*) cannot hold between a same pair of concepts – a usual consistency check for thesauri – and that such a relation cannot hold between two equivalent concepts. An argument is thus considered to attack another if they link the same two concepts with different mapping relations.

Frameworks tested. For our evaluation, we experimented with the following selection of framework and attack strategy settings:

Baseline. This consists of a single aggregation – *union* – of mappers' results into a single set of mappings.

F1 (Strength-based, attacks based on relation disjointness). This setting corresponds to the S-VAF described in Section 2.3 with the NARD attack strategy. Two versions are explored: (F1$_{cont}$) adopting the confidence values produced by the mapper as the strength of the generated arguments; (F1$_{disc}$) applying a threshold (0.5) on the original confidence values to produce arguments with a discrete strength — 0 if the confidence level is below 0.5, 1 otherwise.

F2 (Strength-based, attacks based on absent correspondences). This setting corresponds to an S-VAF with the NAF attack strategy. The same two alternatives as for the previous framework are explored (F2$_{cont}$ and F2$_{disc}$).

F3 (Plurality voting-based, attacks based on absent correspondences). This setting combines the Sup-VAF framework of Section 2.4 with the NAF strategy.

F4 (Borda count-based, attacks based on absent correspondences). This is the Borda count Sup-VAF framework of Section 2.4, applying the NAF strategy.

Mapper configuration. For all settings, three groupings are considered: (1) the three *OAEI* participants; (2) our three *Homegrown* matchers; (3) *All* matchers.

Preference ordering. For all settings, we create an audience for each mapper involved. We define a complete preference order by defining a default order that is adapted, for each audience, by lifting itself to first position: for *OAEI*, the default order is Falcon>Silas>DSSim, but for the Silas audience the order defined is Silas>Falcon>DSSim. The default for *Homegrown* is Co-occurrence>SKOS lexical>Edit-distance. For *All*, it is Falcon>Co-occurrence>SKOS lexical>Edit-distance>Silas> DSSim. This order, even though inspired by observing respective mappers' general performances, remains rather arbitrary. Crucially, it is also fixed: we did not aim at analyzing the influence of this factor in our experiment.

3.5 Results and discussion

Tables 1 and 2 show the results we obtained – *w.r.t.* evaluation measures and amount of obtained annotation translation rules – both for individual matchers and their combinations. For brevity, we show the results of evaluation only when using *all* types of mappings in order to produce rules. We also performed evaluation using only the *exactMatch* ones, but that did not bring significant changes, both for absolute and relative performances of matchers and frameworks.

Mapper	#Rules	P-a	R-a	J-a		Mapper	#Rules	P-a	R-a	J-a
DSSim	9467	13.3	09.4	07.5		SKOS	13207	40.9	43.1	0.29.9
Falcon	3618	52.5	36.6	30.7		Co-occurrence	15742	13.6	79.5	12.7
Silas	9358	45.5	42.6	31.4		Edit distance	20065	31.6	43.5	24.4

Table 1. Individual mappers (P-a, R-a and J-a are expressed as percentages)

		OAEI					Homegrown					All			
Setting		#R	P-a	R-a	J-a		#R	P-a	R-a	J-a		#R	P-a	R-a	J-a
Baseline		16990	32.6	46.8	26.0		37421	13.0	79.8	12.3		45052	12.0	80.0	11.4
F1$_{cont}$		16800	32.6	46.8	26.0		36492	12.8	74.6	12.0		43017	11.6	71.5	10.9
F1$_{disc}$		16799	32.6	46.8	26.0		36332	12.1	70.3	11.3		41222	10.8	66.7	10.2
F2$_{cont}$		829	52.6	07.5	07.2		5021	52.8	37.0	31.3		835	53.3	07.0	06.8
F2$_{disc}$		828	52.6	07.5	07.2		7346	50.0	37.3	31.0		833	53.2	07.0	06.8
F3		2816	53.6	31.5	27.4		11912	41.9	45.3	29.2		26721	07.6	78.8	07.3
F4		16970	32.5	46.6	25.9		37383	13.0	79.6	12.2		836	53.3	07.1	06.9

Table 2. Argumentation on combined mappers (P-a, R-a and J-a are expressed as percentages)

One can first observe the great difference between F1 and F2 – F1 filtering out only a few mappings compared to the baseline. The NARD strategy actually does not result in the generation of many counter-arguments, causing final results similar to those of the union of matchers. This is especially true for OAEI matchers, which output almost only exactMatch mappings – Silas outputs relatedMatch links, but these seem to relate concepts not involved in exactMatch links, even considering Falcon and DSSim. Results vary more for the Homegrown and All combinations, as these include many mappings with different relations, as well as with different strengths, implying more (successful) attacks. Making strengths discrete seems to have muscled up some counter-arguments, leading to slightly stricter (but less efficient!) selection.

F2 is much more selective. When a counter-argument with strength 1 is generated for one matcher, it is likely to defeat the positive arguments issued by matchers with lesser preference. For a given audience, a selective matcher causes the removal, from the subjectively acceptable mappings, of many results from all matchers below him. When each audience privileges the arguments produced by the matcher it represents, this amounts to filter out from the objectively acceptable mappings all those beyond the intersection of mappings with strength

1. This of course implies an expected great increase in precision and a decrease in recall, compared to the union of results. This also makes the practical interest of NAF with such a strength and preference configuration quite low. And it suggests further experiments, with different preference order patterns and default strengths for counter-arguments. For the OAEI combination (as well as for All, which includes it), the intersection is very small (caused by DSSim missing a lot of good mappings) which causes recall to be dramatically low. For the Homegrown configuration, which combines much less stringent mappers, the intersection is larger, explaining an evolution for precision and recall which is more beneficial. Note that there is almost no difference between the continuous and discrete settings for OAEI and All configurations. For these, the OAEI mappings almost entirely dictate the intersection, and most of them already have a strength of 1 – out of Falcon's 3,697 mappings, only 20 have a strength lower than 1. For the Homegrown configuration the effect is opposite to the one obtained for F1: a number of mappings are now "saved", as their strength being discretized up to the one of counter-arguments. However, even if saved mappings are numerous, their consequence on evaluation results is not striking, arguably because of their involving infrequent concepts in the collection. These observations lead to the conclusion that anticipating the effect of making strengths discrete is difficult, without more precise knowledge on the content of alignments.

For OAEI, the severe selection caused by NAF is partly compensated in F4 because of our ranking strategy. Falcon outputs a smaller number of precise results, all of them with a strength of 1. All the good mappings are therefore not attackable: if DSSim produces an attack on one Falcon correspondence, the rank of the attacker is very likely to be lower than the rank of the attacked.

The results for homegrown mappers hint at F3 being the only one able to compensate for attacks on correct correspondences, if enough mappers vote for them. This is certainly true for the OAEI combination, where framework 3 has produced the best precision. This is due the fact that using such framework, it is possible to retrieve significant part of the intersection sets of all mappings, considering the selection of the mappings based on supporters. For example, if both Falcon and DSSim have a positive argument in favour a mapping, independently of the strength of a possible negative argument against the mapping from Silas, the mapping is acceptable. But yet this is not always done at the cost of recall. Even if F3 had worse recall than Silas, it obtains more resulting mappings than F2 with the same continuous setting.[5]

The same applies for the "homegrown" combination. F3 has a slightly lower recall than F2 with continuous strengths, but, again, better precision and Jaccard average than the baseline results, and by an even greater margin. Even when individual mappers return large sets of overlapping mappings, argumentation with voting appears to be more promising than simple union. The results for the last All combination however hint that this positive effect may disappear

[5] Note that our evaluation strategy computes precision on the basis of books for which alignment allows to compute new annotations; it is therefore possible to have a greater set of mappings with a better general precision.

when the number of combined mappers gets bigger, and their precision lower. When too many lax mappers are involved, it is possible that wrong mappings find enough supporters to remain undefeated – the combined influence of DSSim and the un-filtered co-occurrence matcher may be instrumental here.

4 Related work and conclusion

Many methods, such as in [1–3], articulate mappings on a source basis: all mappings from a given source are selected (or weighted, in a weighted sum aggregation system) at once. This can be compared to the preference relation over mapping sources that we use. However, our framework is more precise, since it considers every mapping individually. In this respect, the alignment argumentation frameworks of [9, 5, 6, 8], which we re-use and extend, relate to the efforts focusing on the logical soundness of alignments. As an example, [18, 19] investigates how to detect individual mappings which cause inconsistencies, considering both aligned ontologies and proposed alignments. However, these approaches, similarly to the way argumentation is done in [6], require full-fledged formal ontologies, which will lack in many applications.

Instead, we have experimented with counter-argument generation techniques which can be applied to a wider range of cases. Our proposal to consider the strength of mapping arguments – and the consensus about them – assumes that quantitative aspects of alignment can help to compensate for the lack of formal knowledge, in contexts such as our library case.

However, our results are somehow inconclusive *wrt.* our initial research questions on the benefits of using strengths and consensus in argumentation. In some cases performances are comparable to those of best individual matchers. This is a significant outcome, when the best performing matcher is not known in advance. Still, no framework manages to outperform baseline merging for every configuration. Worse, results point at complex phenomena that may be inherent to combining alignments resulting from very different strategies – confidence assignments, filtering of results... Further investigation is therefore necessary.

First, we will complete our experiments by considering negative arguments based on relation disjointness for the frameworks 3 and 4 and comparing our results with using the basic VAF framework. Beyond, the problem of negative argument generation needs more attention. In our type of application scenarios, we cannot turn to formalized reasoning as done in [6]. It would be still interesting to investigate techniques that take into account more semantic constraints than done in our current strategies, using for instance detection of mapping cycles, or equivalence mappings that relates one concept to two distinct ones. We might benefit here from the constraints specified in the latest SKOS developments [20].

Relevance feedback, as used in [4, 1–3], is also absent in our argumentation system, in which only abstract arguments are considered. A possible option could be to combine both approaches, and raise counter-arguments based on the evaluation – either directly by assessing a correspondence, or in an end-to-end way by studying its effects on the application at hand.

Acknowledgements Authors are supported by the EU Programme Alban for High Level Scholarships for Latin America, the EU eContentPlus project TELplus and the Dutch NWO programme CATCH (STITCH project).

References

1. Tan, H., Lambrix, P.: A method for recommending ontology alignment strategies. In: 6th Intl. Semantic Web Conference (ISWC 2007), Busan, Korea (2007)
2. Ehrig, M., Staab, S., Sure, Y.: Bootstrapping ontology alignment methods with apfel. In: 4th Intl. Semantic Web Conference (ISWC 2005), Galway, Ireland (2005)
3. Mochol, M., Jentzsch, A., Euzenat, J.: Applying an analytic method for matching approach selection. In: Ontology Matching Workshop, ISWC 2006. (2006)
4. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: Start making sense: The chatty web approach for global semantic agreements. J. Web Semantics **1**(1) (2003)
5. dos Santos, C.T., Moraes, M.C., Quaresma, P., Vieira, R.: A cooperative approach for composite ontology mapping. Journal of Data Semantics **10** (2008) 237–263
6. Laera, L., Blacoe, I., Tamma, V., Payne, T.R., Euzenat, J., Bench-Capon, T.: Argumentation over ontology correspondences in mas. In: 6th Intl. Conference on Autonomous Agents and Multi-Agent Systems. (2007)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n–person games. AI **77** (1995)
8. Laera, L., Tamma, V., Payne, T.R., Euzenat, J., Bench-Capon, T.: Reaching agreement over ontology alignments. In: ISWC 2006. (2006)
9. Bench-Capon, T.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation **13** (2003)
10. Isaac, A., van der Meij, L., Schlobach, S., Wang, S.: An empirical study of instance-based ontology matching. In: ISWC 2007, Busan, Korea (2007)
11. dos Santos, C.T., Quaresma, P., Vieira, R.: An extended value-based argumentation framework for ontology mapping with confidence degrees. In: Argumentation in Multi-Agent Systems, 4th Intl. Workshop, Honolulu, HI, USA (2007)
12. Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W.R., Yatskevich, M.: Results of the ontology alignment evaluation initiative 2006. In: Ontology Matching Workshop, ISWC 2006. (2006)
13. de Borda, J.C.: Mémoire sur les elections au scrutin. Histoire de l'Acadmie Royale des Sciences (1781)
14. Hu, W., Zhao, Y., Li, D., Cheng, G., Wu, H., Qu, Y.: Falcon-AO: results for oaei 2007. In: Ontology Matching Workshop, ISWC 2007. (2007)
15. Nagy, M., Vargas-Vera, M., Motta, E.: DSSim – managing uncertainty on the semantic web. In: Ontology Matching Workshop, ISWC 2007. (2007)
16. Ossewaarde, R.: Simple library thesaurus alignment with SILAS. In: Second Intl. Workshop on Ontology Matching, ISWC 2007. (2007)
17. Isaac, A., Matthezing, H., van der Meij, L., Schlobach, S., Wang, S., Zinn, C.: Putting ontology alignment in context: usage scenarios, deployment and evaluation in a library case. In: ESWC 2008, Tenerife, Spain (2008)
18. Stuckenschmidt, H., van Harmelen, F., Serafini, L., Bouquet, P., Giunchiglia, F.: Using c-owl for the alignment and merging of medical ontologies. In: Formal Biomedical Knowledge Representation Workshop, KR 2004, Whistler, Canada (2004)
19. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Applying an analytic method for matching approach selection. In: Ontology Matching Workshop. (2006)
20. Miles, A., Bechhofer, S.: Skos reference. Technical report, W3C (January 25 2008)

A Community Based Approach for Managing Ontology Alignments

Gianluca Correndo, Harith Alani, Paul Smart

University of Southampton,
Electronic and Computer Science Department
[gc3, ha, ps02v]@ecs.soton.ac.uk,
WWW home page:http://ecs.soton.ac.uk
SO17 1BJ, United Kingdom

Abstract. The Semantic Web is rapidly becoming a defacto distributed repository for semantically represented data, thus leveraging on the added on value of the network effect. Various ontology mapping techniques and tools have been devised to facilitate the bridging and integration of distributed data repositories. Nevertheless, ontology mapping can benefit from human supervision to increase accuracy of results. The spread of Web 2.0 approaches demonstrate the possibility of using collaborative techniques for reaching consensus. While a number of prototypes for collaborative ontology construction are being developed, collaborative ontology mapping is not yet well investigated. In this paper, we describe a prototype that combines off-the-shelf ontology mapping tools with social software techniques to enable users to collaborate on mapping ontologies.

1 Introduction

The transformation of the Web from a mere collection of documents to a queryable Knowledge Base (KB) is one of the most prominent targets of Semantic Web (SW) [1]. To help reach this goal, knowledge repositories need to publish semantic representations of their data models to enable other machines to understand and query their content. To this end, much research and development has focused on building tools and capabilities for ontology and KB construction. However, support for distributed teams to remotely and continuously collaborate on building and updating ontologies and knowledge repositories is still underdeveloped.

Defining an ontology for representing data semantics is usually a costly and time consuming task. Furthermore, knowledge evolves over time which adds to maintenance cost. That is why more and more often successful proposals for information sharing involve user's feedback exploiting a network effect. If an ontology is meant to reflect the views of a specific community and support their knowledge sharing tasks, then the community itself should be empowered to express, formalise, share and mantain a set of ontologies for supporting such tasks [2]. Some ontologies need to be agreed upon by the user community, and this agreement process must be supported by tools and methodologies to allow users to express their views and opinions freely.

The rise of social Web 2.0 applications has demonstrated how general Web users can actively contribute and share all sorts of data and information, such as images, videos, bookmarks, opinions, diaries and experiences. Adopting a similar approach on the SW means supporting users to dynamically and collaboratively build ontologies, add semantics to data, discuss and share views and suggestions, etc. Good and colleagues [3] showed how SW users can successfully collaborate to negotiate and build good quality ontologies when provided with a tool that supports such activities. User-contributed content can also be beneficial for engineering ontology mapping activities, most of which rely on automated linguistic and statistical methods that make use of lexicographic clues and structural information but rarely take into account user input [4]. In this paper we describe a prototype and its underlying approach for facilitating gradual ontology mapping by supporting social collaboration and reuse of mapping results. More specifically, our approach allows the following:

- *Alignment of local ontologies to shared ones*: users can align local models, used for bridging data sources, to shared ontologies by using a number of automated ontology mapping tools. These tools are flexibly plugged into our system;
- *Social interaction and collaboration*: users can discuss ontology alignments and propose changes through a number of social services, such as discussion and voting facilities;
- *Reuse of ontology alignment information*: users can add to, and correct, the alignments suggested by automated ontology mapping tools, or suggested by other users. User feedback and mapping information are logged by the system and reused to improve the accuracy of future alignments on similar concepts;

2 Related Work

The need to make explicit and publish the semantics of the data is becoming increasingly central since more information systems are becoming largely decoupled and separately managed. To this end, the vision of the SW is moving towards a scenario where the task of creating and mantaining ontologies, that formalise data semantics, is going to be handed to the community that actually uses them [2]. In accordance with this vision, the models for making data semantics explicit and exchangeable can be the fruit of a collaborative effort by the community members whom will share the responsibility of ontologies creation and maintenance. Such an effort must be supported by tools and methodologies that allow latent models to emerge as a product of a collaborative effort and dialogue.

Our work taps on the intersection of different but overlapping areas in ontology engineering: collaborative construction and management using social networking tools, data web and sharing of ontology fragments. We briefly highlight the main contenders in these areas and elaborate on their relationship with our work.

Historically speaking, investigations into enhancing user knowledge through collaboration and sharing goes back to the early nineties [5]. Ontolingua [6] is an early proposal in this area, which provides some basic support for users to reuse and extend shared ontologies. Another example is the model discussed by Euzenat in [7], where users can build their local ontologies, get them approved by the community, and get support by a discussion protocol which conveys users' rationales for changes in a formal schema. The Semantic Web has taken this approach further by providing the tools and languages to construct networked semantic representational layers to increase understandability, integration, and reuse of information.

The rise of Web 2.0 approaches has then demonstrated the effectiveness and popularity of collaborative knowledge construction and sharing environments that adopted lighter version of ontologies, where the emphasis is put on the easiness of sharing knowledge rather than creating or adopting static formal ontologies [8,9]. Harnessing Web 2.0 features to facilitate the construction, curation, and sharing of knowledge is currently pursued by different communities. Collaborative Protègè [10] was recently developed as an extension to Protègè to support users to edit ontologies collaboratively, by providing them with services for proposing and tracking changes, casting votes, and discussing issues, thus infusing classical ontology editing with a number of popular social interaction features. Another ontology editor with collaborative support is Hozo [11], which focusses on managing ontology modules and their change conflicts. Good and colleagues demonstrated how good quality ontologies can be built quickly in a collaborative fashion[3]. Other approaches use social tagging as the main driver for enacting collaborative lightweight ontology building (e.g [12,13]). Similarly, other tools are focussing on editing instance data, like OntoWiki [14] and DBin [15] which are prime examples of tools for community-driven knowledge creation. Most of the tools listed above focus on supporting users to collaboratively construct ontologies or to collaboratively populate an ontology with instance data. Unlike these tools, however, our proposed system, OntoMediate, extends the collaborative notion to support the task of *ontology mapping*, where users can collaborate and interact to map their existing ontologies and maintain a quality mapping asset within the community. An approach similar to OntoMediate, that addresses ontology mapping within communities, is the Zhadanova and Shvaiko [16] method. The authors proposed to use similarity of user and group profiles as a driver for suggesting ontology alignments reuse. The focus of that work was on building such profiles to personalise reuse of ontology mappings. In OntoMediate, we are exploring the use of collaborative features (discussions, voting, change proposals) to facilitate the curation and reuse of ontological mappings by the community, to facilitate a social and dynamic integration of distributed knowledge bases. The use of collaboration for achieving consensus on terms' semantics is largely justified because of the social nature of ontologies. In order to mediate possibly conflicting concept's description, user feedback must be taken into account and discussion within the community must be fostered. Our approach is novel in the way it addresses the task of aligning ontologies, by extending and enhancing automatic mapping tools with a full community support. In our approach, alignments are seen as a resource, built and shared by a community. The community is able to investigate, argue, and correct the individual mappings, using various supporting services provided in OntoMediate.

3 The OntoMediate Approach

In the OntoMediate[17] project we are studying how social interactions, collaboration and user feedback can be used in a community in order to ease the task of ontology alignment and ontology mapping sharing. Focus of our research is how to ease the integration of data sources using ontologies and ontology alignments in order to provide an agreed semantics to integrated data.

The implemented prototype is a Web application developed with J2EE and AJAX technologies. The system manages OWL ontologies that are parsed using the Jena API[1]. The system has been designed to be extended via its APIs and is composed of three main subsystems:

- Ontologies and datasets manager;
- Ontology alignment environment;
- Social interaction environment.

3.1 Ontologies and Datasets Manager

This part of the system allows users to register (as well as unregister) the datasets they intend to share with the community and the ontologies that describe their data vocabulary. The ontologies that are loaded onto the system, need to be aligned with one or more shared ontologies in order to enable querying of the published data by the community. The system currently supports different storage types for the ontologies and/or datasets:

- *URL*: only the URL is stored and the ontology is accessed (read only) remotely;
- *Cached file*: the ontology file is uploaded to the system and stored in a file server;
- *Jena RDBMS*: the ontology file is uploaded to the system and stored in a relational database using the Jena database back-end;
- *SPARQL endpoint*: the document is remotely accessed using the SPARQL protocol[2].

Once an ontology is registered with the system, the owner (or everyone if the ontology has been shared within the community) can browse it by using a flexible frame-like interface. The ontology browser displays the hierarchy of concepts, as well as detailed information for the focused concept (selected concept). The detailed information includes: labels, superconcepts, subconcepts, equivalent concepts, concept description (from the `rdfs:comment` annotations), properties and their constraints.

[1] http://jena.sourceforge.net

[2] http://www.w3.org/TR/rdf-sparql-protocol/

3.2 Ontology Alignment Environment

The full automation of ontology alignment is not an easy task [18]. The factors that affect the computation and accuracy of ontology alignments are so delicate that we can not afford not to take into account user input as a contributing factor of paramount importance. It is for this reason that, implementing an environment for aligning ontologies, great attention has been made to the usability issues that could affect this task [19].

Our system provides an API for automated ontology alignment tools to be plugged in and also maintains data structures to store parameters needed by a particular tool to execute (e.g. threshold values or available tool options). The API allows for easy integration of new alignment tools, when they become available, by means of wrappers - some tools have been already integrated with our system (e.g. CROSI mapping system [20], INRIA Align [21] and Falcon OA [22]). These tools allow the system to support the alignment task by proposing to the user some initial candidate mappings. The results from different tools can be merged and the decision of which combination of tools to use can be parameterised together with the configuration used to invoke each tool. The merge of results from different tools is achieved by a weighted mean of each contribution and it is implemented as a normal alignment tool plugged into the system (i.e. different merging alghoritms can be coded and plugged in).

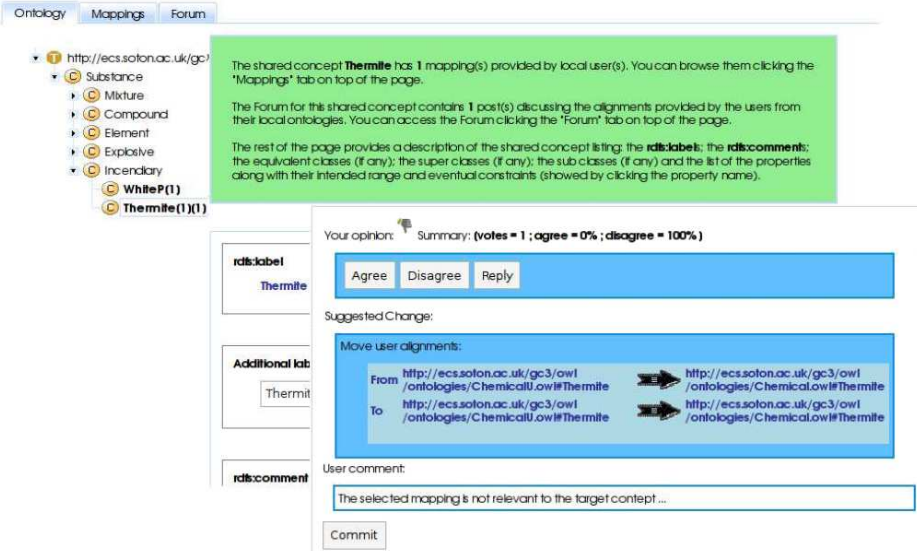
Once the automated mapping has been executed, the results are displayed in a proper interface for reviewing and for searching further alignments. The ontology alignment interface is split into two main panels, the left panel for the source ontology and the right panel for the target ontology, whereas the bottom space is used for summarising the mappings found for the focused source concept. The interface has two view modalities: **Hierarchical** and **Detailed**.

In the **Hierarchical** view the two taxonomies are centered on the source concepts that have been mapped to a target concept, both of which are highlighted. The user can browse both taxonomies and create new mappings by dragging a source concept and dropping it into a destination concept. When the user focusses on a mapping, he/she can switch to a detailed view and the description of the source and target concept are shown side by side.

In the **Detailed** view, the user can map the properties using the same drag & drop facility used for mapping the concepts. The users can also explicitly **reject** some automatically proposed mappings. This choice will be recorded by the system and will be used to filter future mappings towards this target concept, thus increase future ontology alignment *precision*. Alternative interface designs for ontology mapping, such as the one presented in [23], will be considered for future version of the system.

3.3 Social Interaction Environment

This functionality allows users of a community that deal with similar data - and therefore have a mutual interest to maintain good quality alignments - to socially interact with each other. The aim of the social interaction is to exploit community feedback in order to enhance the overall quality of the ontology alignment and achieve agreement on semantics of concepts by means of community acceptance. This subsystem displays to the user three views: **Ontology** view; **Mappings** view and **Forum** view.



The shared concept **Thermite** has **1** mapping(s) provided by local user(s). You can browse them clicking the 'Mappings' tab on top of the page.

The Forum for this shared concept contains **1** post(s) discussing the alignments provided by the users from their local ontologies. You can access the Forum clicking the 'Forum' tab on top of the page.

The rest of the page provides a description of the shared concept listing: the **rdfs:labels**; the **rdfs:comments**; the equivalent classes (if any); the super classes (if any); the sub classes (if any) and the list of the properties along with their intended range and eventual constraints (showed by clicking the property name).

Your opinion: Summary: **(votes = 1 ; agree = 0% ; disagree = 100%)**

Suggested Change:

Move user alignments:

From http://ecs.soton.ac.uk/gc3/owl/ontologies/ChemicalU.owl#Thermite → http://ecs.soton.ac.uk/gc3/owl/ontologies/Chemical.owl#Thermite

To http://ecs.soton.ac.uk/gc3/owl/ontologies/ChemicalU.owl#Thermite → http://ecs.soton.ac.uk/gc3/owl/ontologies/Chemical.owl#Thermite

User comment:

The selected mapping is not relevant to the target contept ...

Fig. 1. Discussion environment - Ontology View - Post

The **Ontology** view (see Figure 1 top-left corner) displays an enhanced taxonomy browser for the selected shared ontology. The enhancements concern the user activities affecting the shared concepts, visualising additional information (e.g. number of incoming mapping per concept are reported in brackets like the number of post exchanged in the forum discussing such mappings). Moreover, the interface allows to inspect the set of labels used for equivalent concepts (i.e. the ones provided with the alignments) in local ontologies (see the *Additional labels* text field in Figure 1). The user or administrator can edit such labels and add them to the shared concept to enrich the concept description with users' contributions. The new mapping, and the edited/added labels, will be logged in a database to be reused later to improve the *recall* of future ontology alignment tasks (section 4.2).

When the user selects a concept that has some user mappings associated with it, he/she can switch to the **Mappings** view that displays information about the local mappings for the focused concept. The user can then inspect a summarised description (i.e. subconcepts, superconcepts, properties etc.) of the local concepts and decide if they are relevant to the focused target concept or initiate a discussion thread in order to change them. The change proposal is composed of a thread post, that describes in natural language the content of the proposal, and a formal description of the operation to discuss. The proposed change can affect a number of alignments and may lead, if the proposal is accepted, to the relocation of such alignments to a different target concept. If the target concept refferenced in the change operation is not yet present in the ontology, a new one will be created within the hierarchy in accordance with the input given by the users in the forum. The possibility to create new concepts to host user alignments provides a way to reshape (even if only by additions) the target ontology in function of the (meta)data provided by users.

The system provides a forum for the discussion of the users' proposals (see Figure 1 bottom-right corner). Every time a user proposes a change using the mappings view, a new thread is created in the forum and other users are free to debate the proposal, **reply** the proposal with a new one or simply **agree** or **disagree** with it. The user's vote is computed for update the proposal statistics (i.e. number of votes, percentage of approvals and disapproval) that is promptly displayed along the proposal.

The new action item associated with a target concept is notified to every interested user by means of RSS feeds whose the interested users can subscribe to. Once a proposal has reached a critical mass (e.g. when the majority of users affected by the change have expressed their opinion) it will be endorsed, or submitted to the administrator in order to judge it and reach a final decision.

4 Working Example

In order to better explain our approach and show how users' feedback can be used in order to improve the ontology matching task, we report on a small example in the chemical domain and the findings of a working experiment. In this example, two users want to share information on hazardous chemical compounds. They each create an ontology that reflect the nature and structure of their data sources (in our example the users deal with data about *Landmines* and *Hazardous Components*, see Table 1).

Table 1. Domain ontologies used in the experiment

Name	**Domain**	**n°Concepts**	**Main Concepts**
		Shared Ontology	
Chemical	Chemistry	130	Element, Compound, Explosive
		Local Ontologies	
Landmine	Explosive devices	830	Country, Explosive Device, Material
Hazardous Components	Hazardous materials and devices	89	Explosive, Flammable, Container

4.1 Alignment task

This tiny community is provided with a shared domain ontology where a set of entities and relationships relevant to the chemical domain is defined (see Table 1). The two users need to align their local ontologies to the shared one in order to exchange information and integrate their data. To fulfill this task, the users use off_the_shelf automatic tools with the **Ontology Alignment** environment (see section 3.2). The automatic ontology alignment tools provide an initial set of alignments that the users can revise, using the system interface explicitly stating the correct alignments and the incorrect ones. With the same interface, the users can then browse the two ontologies and provide manual alignments if required. At the moment only equivalence relation is supported for expressing alignments but the adoption of more expressive primitives is under study. In this scenario the local ontologies act as "contexts" of their respective data sources (following the nomenclature used by Bouquet et al. [24]) while the shared ontology is meant to provide an ontological formalisation of the domain to enable the actual data integration. They are the objects that catalyse the consensus process.

4.2 Reuse of information from mappings

The alignments provided by the alignment task will be reused to improve automatic future alignments toward the same target ontology. Lexical labels from users' ontologies can be adopted by the shared model as *rdfs:label* that can be considered in future automatic alignment tasks in an attempt to improve performance and accuracy of automatic mapping tools. Within the chosen domain (i.e. hazardous chemical compounds, but the assumption holds in other domains), different labels can represent the same concept (e.g. the explosive *HMX* is also known as *Octogen* or *Cyclotetramethylene-tetranitramine*, see Table 2 for a summary of the labels logged from the alignment activity). The working assumption is that, gathering all the labels related to a concept from local representations, and learning which alignments must be avoided in the future (e.g. rejected by users), can help to increase the performance of automated alignments. As an example, assuming the two users of this example have subsequently aligned their ontologies, the labels collected from the first alignment (see Table 2) can be used for improving the performances of the second. Manual mappings discovered by the first user (e.g. *Black Powder* ≡ *Gun Powder* or *Nitromethane* ≡ *Nitrocarbol*) can in fact helping the discovery of target concepts that would be missed otherwise by automatic tools. Such additional user's labels can in fact bring, if integrated in the shared model, to an increase in automated tools precision and recall for subsequent alignments.

4.3 Social interaction

Browsing the definition of the shared ontology, the users can revise each other's alignments to check that the definition of the local concepts is relevant to the targeted shared concept. The self curation of the shared alignments is an important premise of the approach; users that are interested in integrating their data

Table 2. Alignments based on past users activity

Source concept ≡ Target concept	
Discovered by system and proposed to user	
Black Powder ≡ Gun Powder	Black Iron Oxide ≡ Magnetite
Magnesium ≡ Mg	Nitromethane ≡ Nitrocarbol
Red P ≡ Red Phosphorus	White P ≡ White Phosphorus
Learnt from user input to be wrong and rejected	
Red Iron Oxide ≡ Iron Oxide	Nitromethane ≡ Nitroethane

or in querying the integrated knowledge base have a main concern in browsing such alignments, providing feedback and starting corrective operations whenever needed.

Automated ontology alignment tools usually fail to catch the difference among lexically similar concepts such as *Nitromethane* and *Nitroethane*. Despite their lexical and chemical similarity, it is very important to distinguish the two (the first can be used as an explosive while the second can not). For this reason, once a user has found the incorrect alignment (i.e. *Nitromethane ≡ Nitroethane*) inspecting the local concept definition, he/she can select the faulty alignment and initiate a change process. Along with the incorrect mapping, the user can provide the URI of the suggested correct target concept (i.e. *Nitrocarbol*, a synonym of *Nitromethane*) and issue a change proposal. If no suitable concept can be found in the target ontology the user can suggest the creation of a new one providing its location in the targeted hierarchy. The proposal will be posted in the forum dedicated to the maintenance of the shared concept alignment asset. The community can be alerted of the change proposal by RSS feed subscription (every target concept has a feed where new posts are published, and every interested user can register to the feed) and inspect the change proposal, discuss it on the forum, replying to the post or just expressing dis/agreement with the content of such proposal.

4.4 Alignment asset management

Once the two ontologies have been aligned with the shared model, they can be exploited for assuring a meaning preserving information exchange between the components of the community. The discussion fostered in the social environment and the constant supervision by the users upon the ontology alignments help in mantaining agreement and awareness on terms' semantics within the community.

5 Discussion

Collaborative ontology mapping has a great potential in enhancing performance and in sharing results of automatic mapping tools. The system presented in this paper supports users in their ontology mapping activities and logs their feedback to further enhance the output of automated ontology mapping tools. Moreover it provides social features for community driven mapping revisioning and limited support for shared ontology evolution.

Ontology mapping is inherently difficult, and can be influenced by various issues. For example, some mappings can be **user or context dependent**, in which case a mapping that has been approved by some users may not necessarily suit others. **Mapping popularity** can be used to weight each ontology alignment. The degree of popularity of a specific alignment can be taken into account when displaying alignment suggestions to the user. Storing user profiles to **personalise mappings** has been proposed elsewhere [16].

When reusing mapping results, it is important to prevent **error propagation**. It is important to build a user interface in such a way to discourage **blind reuse of mappings**. OntoMediate allows the community to flag, discuss, and democratically change incorrect mappings, but this is of course dependent on users spotting erroneous mappings. If a mapping is reverted, it will be important to readjust its popularity accordingly.

In addition, mappings that receive repeated change proposals or become subject to long and intense discussions may be regarded as **controversial or debatable mappings**. Such mappings may also need to be handled with care when used or reused suggesting administrators to create appropriate ontological description to better characterize those particular local concepts.

OntoMediate uses off the shelf automatic ontology mapping tools, and hence the complexity of its mappings are largely based on those of the mapping tools. The current implementation of OntoMediate allows users to manually map entities expressing simple one to one mapping. More complex mappings, such as mapping a union of classes or linking properties by means of transforming functions, is not currently supported. However, it has been reported that when engineering ontologies collaboratively, complex OWL constructs are often not required [9].

Ontology mapping is a not an easy task, and hence users will not expected to link their ontologies without a clear **added value**. The ultimate goal of OntoMediate is to facilitate distributed querying and integration of knowledge bases in a community. Therefore, in addition to displaying concept mappings, it will be important to also display some information about the knowledge that each mapped ontology brings to the table. Showing what data a specific mapping or a whole ontology is bringing to the community might encourage others to (a) see the general value of this mapping and hence offer their expertise and help to map the new ontology correctly, and (b) map their ontologies to others if they have not already done so (e.g. to link their data to the new repository).

The approach we focused on in OntoMediate is based on a small to medium size community, sharing interests and goals that can benefit from integrating their data. In OntoMediate, it is presumed that an overall administrator can act as the ultimate curator of the system. For such an approach to **scale up to the Web** as a whole, the wisdom of the community will have to be the final ruler. Wikipedia is a fine example of how this can work, and the Linked Data initiative is a first step to creating a wide network of linked semantic data [25]. However, demonstrating added value will be more difficult once the community is too large and diverse, and hence it will probably breakup into sub communities with similar requirements.

6 Summary and Future Work

This paper presented a prototype for supporting ontology mapping with community interactions, where users can collaborate on aligning their ontologies, and manually-driven alignments can be stored and reused later. Our initial experiment showed good potential of increasing both precision and recall in ontology mapping when reusing past mapping results. Next, we plan to run much larger experiments to further test the validity of the approach, and the usability of the services and features that it provides. We have lately implemented services that exploits the managed alignments for translating queries and data. In the near future we will also implement services to allow users to submit *formula* to mediate between concepts or data that might not be directly mappable (e.g. when the concepts are culture-dependent, or when data property values are function of different other values). Additionally, we will next focus on building the capability to allow users to perceive, and query, the integrated KBs, thus increasing added value. The ontology alignments and the social network will be exploited to focus the search task. We will make the system available to the public online in the next few weeks.

7 Acknowledgements

This work was partially funded by a grant awarded to General Dynamics UK Ltd. and the University of Southampton as part of the Data and Information Fusion Defence Technology Centre (DIF DTC) initiative. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the UK Ministry of Defence, or the UK Government.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (May 2001)
2. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. Intelligent Systems, IEEE **21**(3) (2006) 96–101
3. Good, B.M., Tranfield, E.M., Tan, P.C., Shehata, M., Singhera, G.K., Gosselink, J., Okon, E.B., Wilkinson, M.D.: Fast, cheap and out of control: A zero curation model for ontology development. In Altman, R.B., Murray, T., Klein, T.E., Dunker, A.K., Hunter, L., eds.: Pacific Symposium on Biocomputing, World Scientific (August 2006) 128–139
4. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer Verlag (2007)
5. Patil, R., Fikes, R., Patel-Schneider, P.F., McKay, D., Finin, T.W., Gruber, T.R., Neches, R.: The DARPA knowledge sharing effort: A progress report. In: KR. (1992) 777–788
6. Farquhar, A., Fikes, R., Rice, J.: The Ontolingua server: A tool for collaborative ontology construction (1996)
7. Euzenat, J.: Building consensual knowledge bases: Context and architecture. In Mars, N., ed.: Towards Very Large Knowledge Bases - Proceedings of the KB&KS '95 Conference. (1995) 143–155

8. Correndo, G., Alani, H.: Survey of tools for collaborative knowledge construction and sharing. In: Workshop on Collective Intelligence on Semantic Web (CISW 2007). (November 2007)
9. Noy, N., Chugh, A., Alani, H.: The CKC challenge: Exploring tools for collaborative knowledge construction. IEEE Intelligent Systems **Jan/Feb** (2008)
10. Tudorache, T., Noy, N.: Collaborative Protégé. In: Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at WWW 2007, Banff, Canada (2007)
11. Kozaki, K., Sunagawa, E., Kitamura, Y., Mizoguchi, R.: Distributed and collaborative construction of ontologies using hozo. In: Proc. WWW 2007 Workshop on Social and Collaborative Construction of Structured Knowledge, Banff, Canada (May 2007)
12. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: BibSonomy: A social bookmark and publication sharing system. In: Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14^{th} International Conference on Conceptual Structures. (2006)
13. Zacharias, V., Braun, S.: SOBOLEO – social bookmarking and lightweight engineering of ontologies. In: Proc. WWW 2007 Workshop on Social and Collaborative Construction of Structured Knowledge, Banff, Canada (May 2007)
14. Auer, S., Dietzold, S., Lehmann, J., Riechert, T.: OntoWiki: A tool for social, semantic collaboration. In: Workshop on Social and Collaborative Construction of Structured Knowledge (CKC) at WWW 2007, Banff, Canada (2007)
15. Tummarello, G., Morbidoni, C., Nucci, M.: Enabling semantic web communities with DBin: An overview. In: Proc. 5^{th} Int. Semantic Web Conf., ISWC 2006, Athens, GA, USA. (2006)
16. Zhdanova, A.V., Shvaiko, P.: Community-driven ontology matching. In: ESWC. (2006) 34–49
17. Correndo, G., Kalfoglou, Y., Smart, P., Alani, H.: A community based approach for managing ontology alignments. In: 16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW 2008). (2008) to appear.
18. Kalfoglou, Y., Schorlemmer, M., Uschold, M., Sheth, A., Staab, S.: Semantic interoperability and integration. Seminar 04391 - executive summary, Schloss Dagstuhl - International Conference and Research Centre (September 2004)
19. Falconer, S.M., Noy, N.N., Storey, M.A.: Towards understanding the needs of cognitive support for ontology mapping. In: Ontology Matching Workshop. (2006)
20. Kalfoglou, Y., Hu, B., Reynolds, D., Shadbolt, N.: Capturing, representing and operationalising semantic integration (CROSI) project - final report (October 2005)
21. Euzenat, J.: An api for ontology alignment. In: Proc. 3^{rd} Int. Semantic Web Conf. (ISWC), Hiroshima ,Japan (2004)
22. Jian, N., Hu, W., Cheng, G., Qu, Y.: Falcon-AO: Aligning ontologies with falcon. In: Workshop on Integrating Ontologies (K-CAP 2005). (2005) 85–91
23. Falconer, S.M., Storey, M.A.: A cognitive support framework for ontology mapping. In: Proc. of 6^{th} Int. Semantic Web Conf., Busan, Korea. (2007)
24. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: {C-OWL}: Contextualizing ontologies. In Sekara, K., Mylopoulis, J., eds.: Proceedings of the Second International Semantic Web Conference. Number 2870 in Lecture Notes in Computer Science, Springer Verlag (October 2003) 164–179
25. Bizer, C., Cyganiak, R., Heath, T.: How to publish linked data on the web. http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/ (2007)

Results of the
Ontology Alignment Evaluation Initiative 2008 *

Caterina Caracciolo[1], Jérôme Euzenat[2], Laura Hollink[3], Ryutaro Ichise[4], Antoine Isaac[3], Véronique Malaisé[3], Christian Meilicke[5], Juan Pane[6], Pavel Shvaiko[7], Heiner Stuckenschmidt[5], Ondřej Šváb-Zamazal[8], and Vojtěch Svátek[8]

[1] FAO, Roma, Italy
Caterina.Caracciolo@fao.org
[2] INRIA & LIG, Montbonnot, France
jerome.euzenat@inria.fr
[3] Vrije Universiteit Amsterdam, The Netherlands
{laurah,vmalaise,aisaac}@few.vu.nl
[4] National Institute of Informatics, Tokyo, Japan
ichise@nii.ac.jp
[5] University of Mannheim, Mannheim, Germany
{heiner,christian}@informatik.uni-mannheim.de
[6] University of Trento, Povo, Trento, Italy
pane@dit.unitn.it
[7] TasLab, Informatica Trentina, Trento, Italy
pavel.shvaiko@infotn.it
[8] University of Economics, Prague, Czech Republic
{svabo,svatek}@vse.cz

Abstract. Ontology matching consists of finding correspondences between ontology entities. OAEI campaigns aim at comparing ontology matching systems on precisely defined test sets. Test sets can use ontologies of different nature (from expressive OWL ontologies to simple directories) and use different modalities, e.g., blind evaluation, open evaluation, consensus. OAEI-2008 builds over previous campaigns by having 4 tracks with 8 test sets followed by 13 participants. Following the trend of previous years, more participants reach the forefront. The official results of the campaign are those published on the OAEI web site.

1 Introduction

The Ontology Alignment Evaluation Initiative[1] (OAEI) is a coordinated international initiative that organizes the evaluation of the increasing number of ontology matching systems [7]. The main goal of the Ontology Alignment Evaluation Initiative is 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,

* This paper improves on the "First results" initially published in the on-site proceedings of the ISWC workshop on Ontology Matching (OM-2008). The only official results of the campaign, however, are on the OAEI web site.

[1] http://oaei.ontologymatching.org

tool developers can learn and improve their systems. The OAEI campaign provides the evaluation of matching systems on consensus test cases.

Two first events were 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 Semantic Web Conference (ISWC) [18]. Then, unique OAEI campaigns occurred in 2005 at the workshop on Integrating Ontologies held in conjunction with the International Conference on Knowledge Capture (K-Cap) [2], in 2006 at the first Ontology Matching workshop collocated with ISWC [6], and in 2007 at the second Ontology Matching workshop collocated with ISWC+ASWC [8]. Finally, in 2008, OAEI results were presented at the third Ontology Matching workshop collocated with ISWC, in Karlsruhe, Germany[2].

We have continued previous years' trend by having a large variety of test cases that emphasize different aspects of ontology matching. We have kept particular modalities of evaluation for some of these test cases, such as a consensus building workshop.

This paper serves as an introduction to the evaluation campaign of 2008 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 3-10 discuss in turn the settings and the results of each of the test cases. Section 11 overviews lessons learned from the campaign. Finally, Section 12 outlines future plans and Section 13 concludes.

2 General methodology

We first present the test cases proposed this year to OAEI participants. Then we describe the three steps of the OAEI campaign and report on the general execution of the campaign. In particular, we list participants and the tests they considered.

2.1 Tracks and test cases

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

The benchmark track (§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 and 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.

The expressive ontologies track offers ontologies using OWL modeling capabiities:

- **Anatomy: (§4)** The anatomy real world case is about matching the Adult Mouse Anatomy (2744 classes) and the NCI Thesaurus (3304 classes) describing the human anatomy.

[2] http://om2008.ontologymatching.org

FAO (§5): The FAO test case is a real-life case aiming at matching OWL ontologies developed by the Food and Agriculture Organization of the United Nations (FAO) related to the fisheries domain.

The directories and thesauri track proposed web directories, thesauri and generally less expressive resources:

Directory (§6): The directory real world case consists of matching web sites directories (like open directory or Yahoo's). It is more than 4 thousand elementary tests.

Multilingual directories (§7): The mldirectory real world case consists of matching web site directories (such as Google, Lycos and Yahoo's) in different languages, e.g., English and Japanese. Data sets are excerpts of directories that contain approximately one thousand categories.

Library (§8): Two SKOS thesauri about books have to be matched using relations from the SKOS Mapping vocabulary. Samples of the results are evaluated by domain experts. In addition, we run application dependent evaluation.

Very large crosslingual resources (§9): This real world test case requires matching very large resources (vlcr) available on the web, viz. DBPedia, WordNet and the Dutch audiovisual archive (GTAA), DBPedia is multilingual and GTAA is in Dutch.

The conference track and consensus workshop (§10): Participants were 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 alignments as well as in interesting individual correspondences ("nuggets"), aggregated statistical observations and/or implicit design patterns. Organizers of this track offered diverse a priori and a posteriori evaluation of results. For a selected sample of correspondences, consensus was sought at the workshop and the process was tracked and recorded.

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

test	formalism	relations	confidence	modalities	language
benchmark	OWL	=	[0 1]	open	EN
anatomy	OWL	=	[0 1]	blind	EN
fao	OWL	=	1	expert	EN+ES+FR
directory	OWL	=	1	blind	EN
mldirectory	OWL	=	1	blind	EN+JP
library	SKOS, OWL	narrow-, exact-,	1	blind	EN+DU
vlcr	SKOS, OWL	broad-, relatedMatch	1	blind	EN+DU
conference	OWL-DL	=, $\leq$	[0 1]	blind+consensual	EN

Table 1. Characteristics of test cases (open evaluation is made with already published reference alignments, blind evaluation is made by organizers from reference alignments unknown to the participants, consensual evaluation is obtained by reaching consensus over the found results).

2.2 Preparatory phase

Ontologies to be matched and (where applicable) alignments have been provided in advance during the period between May 19th and June 15th, 2008. 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 ensure that the delivered tests make sense to the participants. The final test base was released on July 1st. The data sets did not evolve after this period.

2.3 Execution phase

During the execution phase, 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 matched 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.

In most cases, ontologies are described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in the Alignment format expressed in RDF/XML [5]. Participants also provided the papers that are published hereafter and a link to their systems and their configuration parameters.

2.4 Evaluation phase

The organizers have evaluated the alignments provided by the participants and returned comparisons on these results.

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 1st. In the case of blind tests only the organizers did the evaluation with regard to the withheld reference alignments.

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 the 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 as well as measures compensating for the lack of complete reference alignments.

In addition, the Library test case featured an application-specific evaluation and a consensus workshop has been held for evaluating particular correspondences.

2.5 Comments on the execution

This year, for the first time, we had less participants than in the previous year (though still more than in 2006): 4 in 2004, 7 in 2005, 10 in 2006, 18 in 2007, and 13 in 2008. However, participants were able to enter nearly as many individual tasks as last year: 48 against 50.

We have had not enough time to systematically validate the results which had been provided by the participants, but we run a few systems and we scrutinized some of the results.

We summarize the list of participants in Table 2. Similar to previous years not all participants provided results for all tests. They usually did those which are easier to run, such as benchmark, directory and conference. The variety of tests and the short time given to provide results have certainly prevented participants from considering more tests.

There is an even distribution of systems on tests (unlike last year when there were two groups of systems depending on the size of the ontologies). This years' participation seems to be weakly correlated with the fact that a test has been offered before.

Software	confidence	benchmark	anatomy	fao	directory	mldirectory	library	vlcr	conference
Anchor-Flood		√	√						
AROMA	√	√	√	√					
ASMOV	√	√	√	√	√				√
CIDER	√	√			√				
DSSim	√	√	√	√	√	√	√	√	√
GeRoMe		√							
Lily	√	√	√	√	√	√	√		√
MapPSO		√		√	√	√			
RiMOM	√	√	√	√	√	√			
SAMBO	√	√	√	√					
SAMBOdtf	√	√	√	√					
SPIDER	√	√							
TaxoMap		√	√		√		√		
Total=13		13	9	8	7	4	3	1	3

Table 2. Participants and the state of their submissions. Confidence stands for the type of result returned by a system: it is ticked when the confidence has been measured as non boolean value.

This year we can still regret to have not enough time for performing tests and evaluations. This may explain why even participants with good results last year did not participate this year. The summary of the results track by track is provided in the following seven 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 ontologies are described in OWL-DL and serialized in the RDF/XML format. The reference ontology is that of test #101. It contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals. Participants have to match this reference ontology with the variations. Variations are focused on the characterization of the behavior of the tools rather than having them compete on real-life problems. They are organized in three groups:

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) obtained by discarding features from some reference ontology. It aims at evaluating how an algorithm behaves when a particular type of 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* that can be suppressed, expanded or flattened;
- *Instances* that can be suppressed;
- *Properties* that can be suppressed or having the restrictions on classes discarded;
- *Classes* that can be expanded, i.e., replaced by several classes or flattened.

Four real-life ontologies of bibliographic references (3xx) found on the web and left mostly untouched (there were added xmlns and xml:base attributes).

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, whose test numbering it (almost) fully preserves.

After remarks of last year we made two changes on the tests this year:

- tests #249 and 253 still had instances in the ontologies, these have been suppressed this year. Hence the test is more difficult than previous years;

– tests which scrambled all labels within the ontology (#201-202, 248-254 and 257-262), have been complemented by tests which respectively only scramble 20%, 40%, 60% and 80% of the labels. Globally, this makes the tests easier to solve.

The kind of expected alignments is still limited: they only match named classes and properties, they mostly use the "=" relation with confidence of 1. Full description of these tests can be found on the OAEI web site.

3.2 Results

All the 13 systems participated in the benchmark track of this year's campaign. Table 3 provides the consolidated results, by groups of tests. We display the results of participants as well as those given by some simple edit distance algorithm on labels (edna). The computed values are real precision and recall and not an average of precision and recall. The full results are on the OAEI web site.

Results in Table 3 show already that the three systems, which last year were leading, are still relatively ahead (ASMOV, Lily and RiMOM) with three close followers (AROMA, DSSim, and Anchor-Flood replacing Falcon, Prior+ and OLA_2 last year). No system had strictly lower performance than edna. Each algorithm has its best score with the 1xx test series. There is no particular order between the two other series.

This year again, the apparently best algorithms provided their results with confidence measures. It is thus possible to draw precision/recall graphs in order to compare them. We provide in Figure 1 the precision and recall graphs of this year. They are only relevant for the results of participants who provided confidence measures different from 1 or 0 (see Table 2). 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). They do not feature the curves of previous years since the test sets have been changed.

These results and those displayed in Figure 2 single out the same group of systems, ASMOV, Lily, and RiMOM which seem to perform these tests at the highest level of quality. So this confirms the leadership that we observed on raw results.

Like the two previous years, there is a gap between these systems and their followers. The gap between these systems and the next ones (AROMA, DSSim, and Anchor-Flood) has reformed. It was filled last year by Falcon, OLA_2, and Prior+ which did not participate this year.

We have also compared the results of this year's systems with the results of the previous years on the basis of 2004 tests, see Table 4. The two best systems on this basis are the same: ASMOV and Lily. Their results are very comparable but never identical to the results provided in the previous years by RiMOM (2006) and Falcon (2005).

system	refalign		edna		Aflood		AROMA		ASMOV		CIDER		DSSim		GeRoMe		Lily		MapPSO		RiMOM		SAMBO		SAMBOdtf		SPIDER		TaxoMap	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
2008																														
1xx	1.00	1.00	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00	0.96	0.79	1.00	1.00	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	0.34
2xx	1.00	1.00	0.41	0.56	0.96	0.69	0.96	0.70	0.95	0.85	0.97	0.57	0.97	0.64	0.56	0.52	0.97	0.86	0.48	0.53	0.96	0.82	0.98	0.54	0.98	0.56	0.97	0.57	0.95	0.21
3xx	1.00	1.00	0.47	0.82	0.95	0.66	0.82	0.71	0.81	0.77	0.90	0.75	0.90	0.71	0.61	0.40	0.87	0.81	0.49	0.25	0.80	0.81	0.95	0.80	0.91	0.81	0.15	0.81	0.92	0.21
H-mean	1.00	1.00	0.43	0.59	0.97	0.71	0.95	0.70	0.95	0.86	0.97	0.62	0.97	0.67	0.60	0.58	0.97	0.88	0.51	0.54	0.96	0.84	0.99	0.58	0.98	0.59	0.81	0.63	0.91	0.22
Symmetric relaxed measures																														
H-mean	1.00	1.00	0.73	1.00	1.00	0.72	error		0.99	0.90	error		error		error		0.99	0.89	error		error		0.99	0.58	0.99	0.59	error		1.00	0.24
2007																														
1xx	1.00	1.00	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00	0.96	0.79	1.00	1.00	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	0.34
2xx	1.00	1.00	0.41	0.56	0.96	0.69	0.96	0.70	0.95	0.85	0.97	0.57	0.97	0.64	0.56	0.52	0.97	0.86	0.48	0.53	0.96	0.82	0.98	0.54	0.98	0.56	0.97	0.57	0.95	0.21
3xx	1.00	1.00	0.47	0.82	0.95	0.66	0.82	0.71	0.81	0.77	0.90	0.75	0.90	0.71	0.61	0.40	0.87	0.81	0.49	0.25	0.80	0.81	0.95	0.80	0.91	0.81	0.15	0.81	0.92	0.21
H-mean	1.00	1.00	0.45	0.61	0.97	0.71	0.96	0.72	0.95	0.85	0.97	0.62	0.97	0.68	0.59	0.54	0.96	0.87	0.52	0.55	0.95	0.83	0.98	0.59	0.98	0.61	0.67	0.62	0.95	0.22

Table 3. Means of results obtained by participants on the benchmark test case (corresponding to harmonic means). The symmetric relaxed measure corresponds to the three relaxed precision and recall measure of [4]. The 2007 subtable corresponds to the results obtained on the results of 2007 tests only (suppressing the 20-40-60-80% alteration).

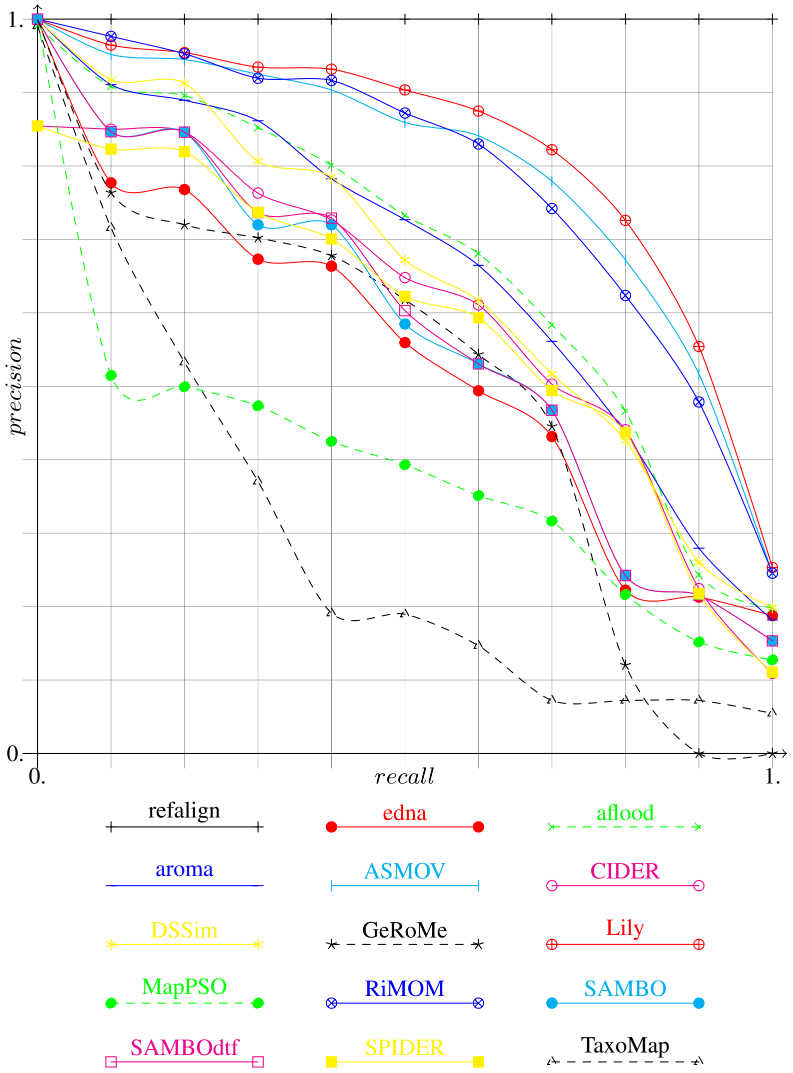


Fig. 1. Precision/recall graphs. They cut the results given by the participants under a threshold necessary for achieving $n\%$ recall and compute the corresponding precision. Systems for which these graphs are not meaningful (because they did not provide graded confidence values) are drawn in dashed lines. This is, as expected, those which have the lower results in these curves.

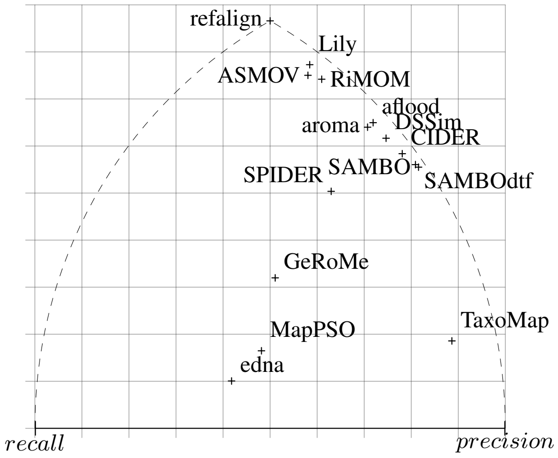


Fig. 2. Each point expresses the position of a system with regard to precision and recall.

Year	2004				2005		2006		2007				2008			
System	Fujitsu		PromptDiff		Falcon		RiMOM		ASMOV		Lily		ASMOV		Lily	
test	Prec.	Rec.	Prec.	Rec.	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	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	0.99	0.99	1.00	0.98	0.99	0.98	0.99	0.98
3xx	0.60	0.72	0.93	0.74	0.93	0.83	0.83	0.82	0.85	0.82	0.81	0.80	0.81	0.77	0.87	0.81
H-means	0.88	0.85	0.98	0.77	0.97	0.96	0.97	0.96	0.97	0.97	0.97	0.96	0.97	0.96	0.98	0.96

Table 4. Evolution of the best scores over the years on the basis of 2004 tests (RiMOM had very similar results to ASMOV's).

4 Anatomy

The focus of the anatomy track is to confront existing matching technology with real world ontologies. Currently, we find such real world cases primarily in the biomedical domain, where a significant number of ontologies have been built covering different aspects of medical research.[3] Manually generating alignments between these ontologies requires an enormous effort by highly specialized domain experts. Supporting these experts by automatically providing correspondence proposals is challenging, due to the complexity and the specialized vocabulary of the domain.

4.1 Test Data and Experimental Setting

The ontologies of the anatomy track are the NCI Thesaurus describing the human anatomy, published by the National Cancer Institute (NCI)[4], and the Adult Mouse Anatomical Dictionary[5], which has been developed as part of the Mouse Gene Expression Database project. Both resources are part of the Open Biomedical Ontologies (OBO). A more detailed description of the characteristics of the data set has already been given in the context of OAEI 2007 [8].

Due to the harmonization of the ontologies applied in the process of generating a reference alignment, a high number of rather trivial correspondences can be found by simple string comparison techniques. At the same time, we have a good share of non-trivial correspondences that require a careful analysis and sometimes also medical background knowledge. The construction of the reference alignment has been described in [3]. To better understand the occurrence of non-trivial correspondences in alignment results, we implemented a straightforward matching tool that compares normalized concept labels. This trivial matcher generates for all pairs of concepts $\langle C, D\rangle$ a correspondence if and only if the normalized label of C is identical to the normalized label of D. In general we expect an alignment generated by this approach to be highly precise while recall will be relatively low. With respect to our matching task we measured approximately 98% precision and 61% recall. Notice that the value for recall is relatively high, which is partially caused by the harmonization process mentioned above. In 2007 we assumed that most matching systems would easily find the trivial correspondences. To our suprise this assumption has not been verified. Therefore, we applied again the additional measure referred to as $recall+$. $recall+$ measures how many non trivial correct correspondences can be found in an alignment M. Given reference alignment R and alignment S generated by the naive string equality matching, $recall+$ is defined as $recall+ = |(R \cap M) - S| \,/\, |R - S|$.

We divided the task of automatically generating an alignment into four subtasks. Task #1 is obligatory for participants of the anatomy track, while task #2, #3 and #4 are optional tasks. Compared to 2007 we also introduced #4 as challenging fourth subtask. For task #1 the matching system has to be applied with standard settings to obtain a result that is as good as possible with respect to the expected F-measure. In particular,

[3] A large collection can be found at `http://www.obofoundry.org/`.

[4] `http://www.cancer.gov/cancerinfo/terminologyresources/`

[5] `http://www.informatics.jax.org/searches/AMA_form.shtml`

we are interested in how far matching systems improved their results compared to last years evaluation. For task #2 an alignment with increased precision has to be found. Contrary to this, in task #3 an alignment with increased recall has to be generated. We believe that systems configurable with respect to these requirements will be much more useful in concrete scenarios compared to static systems. While we expect most systems to solve the first three tasks, we expect only few systems to solve task #4. For this task a part of the reference alignment is available as additional input. In task #4 we tried to simulate the following scenario. Suppose that a group of domain experts already created an incomplete reference alignment by manually validating a set of automatically generated correspondences. As a result a partial reference alignment, in the following referred to as R_p, is available. Given both ontologies as well as R_p, a matching system should be able to exploit the additional information encoded in R_p. We constructed R_p as the union of the correct trivial correspondences and a small set of 54 non trivial correspondences. Thus R_p consists of 988 correspondences, while the complete reference alignment R contains 1523 correspondences.

4.2 Results

In total, nine systems participated in the anatomy task (in 2007 there were 11 participants). These systems can be divided into a group of systems using biomedical background knowledge and a group of systems that do not exploit domain specific background knowledge. SAMBO and ASMOV belong to the first group, while the other systems belong to the second group. Both SAMBO and ASMOV make use of UMLS, but differ in the way they exploit this additional knowledge. Table 5 gives an overview of participating systems. In 2007 we observed that systems of the first group have a significant advantage of finding non trivial correspondences, in particular the best three systems (AOAS, SAMBO, and ASMOV) made use of background knowledge. We will later see whether this assumption could be verified with respect to 2008 submissions.

Compliance measures for task #1 Table 5 lists the results of the participants in descending order with respect to the achieved F-measure. In the first row we find the SAMBO system followed by its extension SAMBOdtf. SAMBO has achieved slightly better results for both precision and recall in 2008 compared to 2007. SAMBO now nearly reaches the F-measure 0.868 which AOAS achieved 2007. This is a notable result, since SAMBO is originally designed to generate alignment suggestions that are afterwards presented to a human evaluator in an interactive fashion. While SAMBO and SAMBOdtf make extensive use of biomedical background knowledge, the RiMOM matching system is mainly based on computing label edit-distances combined with similarity propagation strategies. Due to a major improvement of the RiMOM results, RiMOM is now one of the top matching systems for the anatomy track even though it does not make use of any specific background knowledge. Notice also that RiMOM solves the matching task in a very efficient way. Nearly all matching systems participating 2007 improved their results, while ASMOV and TaxoMap obtained slightly worse results. Further considerations have to clarify the reasons for this decline.

Task #2 and #3 As explained above these subtasks show in how far matching systems can be configured towards a trade-off between precision and recall. To our surprise only four participants submitted results for task #2 and #3 showing that they were able to

System	Runtime	BK	Precision	Recall	Recall+	F-Measure
SAMBO	≈ 12h	yes	$0.869_{0.845}$	$0.836_{0.797}$	$0.586_{0.601}$	$0.852_{0.821}$
SAMBOdtf	≈ 17h	yes	0.831	0.833	0.579	0.832
RiMOM	≈ 24min	no	$0.929_{0.377}$	$0.735_{0.668}$	$0.350_{0.404}$	$0.821_{0.482}$
aflood	1min 5s	no	0.874	0.682	0.275	0.766
Label Eq.	-	*no*	*$0.981_{0.981}$*	*$0.613_{0.613}$*	*$0.000_{0.000}$*	*$0.755_{0.755}$*
Lily	≈ 3h 20min	no	$0.796_{0.481}$	$0.693_{0.567}$	$0.470_{0.387}$	$0.741_{0.520}$
ASMOV	≈ 3h 50min	yes	$0.787_{0.802}$	$0.652_{0.711}$	$0.246_{0.280}$	$0.713_{0.754}$
AROMA	3min 50s	no	0.803	0.560	0.302	0.660
DSSim	≈ 17min	no	$0.616_{0.208}$	$0.624_{0.189}$	$0.170_{0.070}$	$0.620_{0.198}$
TaxoMap	≈ 25min	no	$0.460_{0.586}$	$0.764_{0.700}$	$0.470_{0.234}$	$0.574_{0.638}$

Table 5. Runtime, use of domain specific background knowledge (BK), precision, recall, recall+ and F-measure for task #1. Results of 2007 evaluation are presented in smaller font if available. Notice that the measurements of 2007 have been slightly corrected due to some minor modifications of the reference alignment.

adapt their system for different scenarios of application. These systems were RiMOM, Lily, ASMOV, and DSSim. A more detailed discussion of their results with respect to task #2 and #3 can be found on the OAEI anatomy track webpage[6].

System	Δ-Precision	Δ-Recall	Δ-F-Measure
SAMBO	$+0.024_{0.636\to0.660}$	$-0.002_{0.626\to0.624}$	$+0.011_{0.631\to0.642}$
SAMBOdtf	$+0.040_{0.563\to0.603}$	$+0.008_{0.622\to0.630}$	$+0.025_{0.591\to0.616}$
ASMOV	$+0.063_{0.339\to0.402}$	$-0.004_{0.258\to0.254}$	$+0.019_{0.293\to0.312}$
RiMOM	$+0.012_{0.700\to0.712}$	$+0.000_{0.370\to0.370}$	$+0.003_{0.484\to0.487}$

Table 6. Changes in precision, recall and F-measure based on comparing $M_1 \setminus R_p$ resp. $M_4 \setminus R_p$ with the unknown part of the reference alignment $R \setminus R_p$.

Task #4 Four systems participated in task #4. These systems were SAMBO and SAMBOdtf, RiMOM, and ASMOV. In the following we refer to an alignment generated for task #1 resp. #4 as M_1 resp. M_4. Notice first of all that a direct comparison between M_1 and M_4 is not appropriate to measure the improvement that results from exploiting R_p. We thus have to compare $M_1 \setminus R_p$ resp. $M_4 \setminus R_p$ with the unknown subset of the reference alignment $R_u = R \setminus R_p$. The differences between M_1 (partial reference alignment not available) and M_4 (partial reference alignment given) are presented in Table 6. All participants slightly increased the overall quality of the generated alignments with respect to the unknown part of the reference alignment. SAMBOdtf and ASMOV exploited the partial reference alignment in the most effective way. The measured im-

[6] `http://webrum.uni-mannheim.de/math/lski/anatomy08/`

provement seems to be only minor at first sight, but notice that all of the correspodences in R_u are non trivial due to our choice of the partial reference alignment. The improvement is primarily based on generating an alignment with increased precision. ASMOV for example increases its precision from 0.339 to 0.402. Only SAMBOdtf also profits from the partial reference alignment by a slightly increased recall. Obviously, the partial reference alignment is mainly used in the context of a strategy which filters out incorrect correspondences.

Runtime Even though the submitted alignments have been generated on different machines, we believe that the runtimes provided by participants are nevertheless useful and provide a basis for an approximate comparison. For the two fastest systems, namely aflood and AROMA, runtimes have been measured by the track organizers on the same machine (Pentium D 3.4GHz, 2GB RAM) additionally. Compared to last years competition we observe that systems with a high runtime managed to decrease the runtime of their system significantly, e.g. Lily and ASMOV. Amongst all systems AROMA and aflood, both participating for the first time, performed best with respect to runtime efficiency. In particular, the aflood system achieves results of high quality in a very efficient way.

4.3 Conclusions

In last years evaluation, we concluded that the use of domain related background knowledge is a crucial point in matching biomedical ontologies. This observation is supported by the claims made by other researchers [1, 15]. The current results partially support this claim, in particular the good results of the SAMBO system. Nevertheless, the results of RiMOM and Lily indicate that matching systems are able to detect non trivial correspondences even though they do not rely on background knowledge. To support this claim we computed the union of the alignments generated by RiMOM and Lily. As a result we measured that 61% of all non trivial correspondences are included in the resulting alignment. Thus, there seems to be a significant potential of exploiting knowledge encoded in the ontologies. A combination of both approaches might result in a hybrid matching strategy that uses both background knowledge and the internal knowledge to its full extent.

5 FAO

The Food and Agriculture Organization of the United Nations (FAO) collects large amounts of data about all areas related to food production and consumption, including statistical data, e.g., time series, and textual documents, e.g., scientific papers, white papers, project reports. For the effective storage and retrieval of these data sets, controlled vocabularies of various types (in particular, thesuri and metadata hierarchies) have extensively been used. Currently, this data is being converted into ontologies for the purpose of enabling connection between data sets otherwise isolated from one another. The FAO test case aims at exploring the possibilities of establishing alignments between some of the ontologies traditionally available. We chose a representative subset of them, that we describe below.

5.1 Test set

The FAO task involves the three following ontologies:

- AGROVOC[7] is a thesaurus about all matters of interest for FAO, it has been translated into an OWL ontology as a hierarchy of classes, where each class corresponds to an entry in the thesaurus. For technical reasons, each class is associated with an instance with the same name. Given the size and the coverage of AGROVOC, we selected only the branches of it that have some overlap with the other considered ontologies. We then selected the fragments of AGROVOC about "organisms," "vehicles" (including vessels), and "fishing gears".
- ASFA[8] is a thesaurus specifically dedicated to aquatic sciences and fisheries. In its OWL translation, descriptors and non-descriptors are modeled as classes, so the ontology does not contain any instance. The tree structure of ASFA is relatively flat, with most concepts not having subclasses, and a maximum depth of 4 levels. Concepts have associated annotations, each of which containing the English definition of the term.
- Two specific fisheries ontologies in OWL[9], that model coding systems for commodities and species, used as metadata for statistical time series. These ontologies have a fairly simple class structure, e.g., the species ontologies has one top class and four subclasses, but a large number of instances. They contain instances in up to 3 languages (English, French and Spanish).

Based on these ontologies, participats were asked to establish alignments between:

1. AGROVOC and ASFA (from now on called agrasfa),
2. AGROVOC and fisheries ontology about biological species (called agrobio),
3. the two ontologies about biological species and commodities (called fishbio).

Given the structure of the ontologies described above, the expectation about the resulting alignments was that the alignment between AGROVOC and ASFA (agrasfa) would be at the class level, since both model entries of the thesaurus as classes. Analogously, both the alignment between AGROVOC and biological species (agrobio), and the alignment between the two fisheries ontologies (fishbio) is expected to be at the instance level. However, no strict instructions were given to participants about the exact type of alignment expected, as one of the goals of the experiment was to find how automatic systems can deal with a real-life situation, when the ontologies given are designed according to different models and have little or no documentation.

The equivalence correspondences requested for the agrasfa and agrobio subtracks are plausible, given the similar nature of the two resources (thesauri used for human indexing, with some overlap in the domain covered). In the case of the fishbio subtrack this is not true, as the two ontologies involved are about two domains that are disjoint, although related, i.e., commodities and fish species. The relation between the two domains is that a specific species (or more than one) are the primary source of the goods

[7] `http://www.fao.org/aims/ag_intro.htm`
[8] `http://www.fao.org/fishery/asfa/8`
[9] `http://www.fao.org/aims/neon.jsp`

sold, i.e. the commodity. Their relation then is not an equivalence relation but can rather be seen, in OWL terminology, as an object property with domain and range sitting in different ontologies. The intent of the subtrack fishbio is then to explore the possibility of using the machinery available for inferring equivalence correspondence to non conventional cases.

5.2 Evaluation procedure

All participants but one, Aroma, returned equivalence correspondence only. The non-equivalence correspondences of Aroma were ignored.

A reference alignment was obtained by randomly selecting a specific number of correspondences from each system and then pooling together. This provided a sample alignment A^0.

This sample alignment was evaluated by FAO experts for correctness. This provided a partial reference alignment R^0. We had two assessors: one specialized in thesauri and daily working with AGROVOC (assessing the alignments of the track agrasfa) and one specialized in fisheries data (assessing subtracks agrobio and fishbio). Given the differences between the ontologies, some transformations had to be made in order to present data to the assessors in a user-friendly manner. For example, in the case of AGROVOC, evaluators were given the English labels together with all available "used for" terms (according to the thesauri terminology familiar to the assessor).

dataset	retrieved (A^*)	evaluated (A^0)	correct (R^0)	(A^0/A^*)	(R^0/A^0)
agrasfa	2588	506	226	.19	.45
agrobio	742	264	156	.36	.59
fishbio	1013	346	131	.26	.38
TOTAL	4343	1116	513	.26	.46

Table 7. Size of returned results and samples.

Table 7 summarizes the sample size per each data sets. The second column (retrieved) contains the total number of distinct correspondences provided by all participants for each track. The third column (evaluated) reports the size of the sample extracted for manual assessment. The forth column (correct) reports the number of correspondences found correct by the assessors.

After manual evaluation, we realized that some participants did not use the correct URI in the agrasfa dataset, so some correspondences were considered as different even though they were actually the same. However, this happened only in very few cases.

For each system, precision was computed on the basis of the subset of alignments that were manually assessed, i.e., $A \cap A^0$. Hence,

$$P^0(A, R^0) = P(A \cap A^0, R^0) = |A \cap R^0|/|A \cap A^0|$$

The same was considered for recall which was computed with respect to the total number of correct correspondences per subtrack, as assessed by the human assessors. Hence,

$$R^0(A, R^0) = R(A \cap A^0, R^0) = |A \cap R^0|/|R^0|$$

Recall is expected to be higher than actual recall because it is based only on correspondences that at least one system returned, leaving aside those that no system were able to return.

We call these two measures relative precision and recall because they are relative to the sample that has been extracted.

5.3 Results

Table 8 summarizes the precision and (relative) recall values of all systems, by subtracks. The third column reports the total number of correspondences returned by each system per subtrack. All non-equivalence correspondences were discarded, but this only happened for one systems (Aroma). The fourth column reports the number of alignments from the system that were evaluated, while the fifth column reports the number of correct alignments as judged by the assessors. Finally, the sixth and seventh columns report the values of relative precision and recall computed as described above.

System	subtrack	retrieved $\|A\|$	evaluated $\|A \cap A^0\|$	correct $\|A \cap R^0\|$	RPrecision $P^0(A, R^0)$	RRecall $R^0(A, R^0)$
Aroma	agrasfa	195	144	90	0.62	0.40
	agrobio	2	4	0		
	fishbio	11				
ASMOV	agrafsa	1				
	agrobio	0				
	fishbio	5				
DSSim	agrasfa	218	129	70	0.54	0.31
	agrobio	339	214	151	0.71	0.97
	fishbio	243	166	79	0.48	0.60
Lily	agrasfa	390	105	91	0.87	0.40
MapPSO	agrobio*	6				
	fishbio*	16				
RiMOM	agrasfa	743	194	158	0.81	0.70
	agrobio	395	219	149	0.68	0.95
	fishbio	738	217	118	0.54	0.90
SAMBO	agrasfa	389	176	121	0.69	0.53
SAMBOdtf	agrasfa	650	219	124	0.57	0.55

Table 8. Participant results per datasets. The star (*) next to a system marks those systems which matched properties.

One system (MapPSO) returned alignments of properties, which were discarded and therefore no evaluation is provided in the table. The results of ASMOV were also not evaluated because too few to be considered. Finally, the evaluation of Aroma is incomplete due to the non equivalence correspondence returned, that were discarded before pooling the results together to create the reference alingment.

5.4 Discussion

The sampling method that has been used is certainly not perfect. In particular, it did not allow to evaluate two systems which returned few results (ASMOV and MapPSO). However, the results returned by these system were not likely to provide good recall.

Moreover, the very concise instructions and the particular character of the test sets, clearly puzzled participants and their systems. As a consequence, the results may not be as good as if the systems were applied to polished tests with easily comparable data sets. This provides a honest insight of what these systems would do when confronted with these ontologies on the web. In that respects, the results are not bad.

From DSSim and RiMOM results, it seems that fishbio is the most difficult task in terms of precision and agrasfa the most difficult in terms of recall (for most of the systems). The fact that only two systems returned usable results for agrobio and fishbio makes comparison of systems very difficult at this stage. However, it seems that RiMOM is the one that provided the best results. RiMOM is especially interesting in this real-life case, as it performed well both when an alignment between classes and an alignment between instances is appropriate. Given the fact that in real-life situations it is rather common to have ontologies with a relatively simple class structure and a very large population of instances, this is encouraging.

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 directory matching task was constructed from Google, Yahoo and Looksmart web directories following the methodology described in [9]. The data set is presented as taxonomies where the nodes of the web directories are modeled as classes and classification relation connecting the nodes is 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 matching task which is very large (Google and Yahoo directories have up to $3 * 10^5$ nodes each: this means that the human annotators need to consider up to $(3*10^5)^2 = 9*10^{10}$ correspondences), it uses semi automatic pruning techniques in order to significantly reduce the search space. For example, for the data set described in [9], human annotators consider only 2265 correspondences instead of the full matching problem.

The specific characteristics of the data set are:

- More than 4.500 node matching tasks, where each node matching task is composed from the paths to root of the nodes in the web directories.
- Reference correspondences for all the matching tasks.
- Simple relationships, in particular, web directories contain only one type of relationships, which is the so-called classification relation.
- Vague terminology and modeling principles, thus, the matching tasks incorporate the typical real world modeling and terminological errors.

6.2 Results

In OAEI-2008, 7 out of 13 matching systems participated on the web directories test case, while in OAEI-2007, 9 out of 18, in OAEI-2006, 7 out of 10, and in OAEI-2005, 7 out of 7 did it.

Precision, recall and F-measure results of the systems are shown in Figure 3. These indicators have been computed following the TaxMe2 [9] methodology, with the help of Alignment API [5], version 3.4.

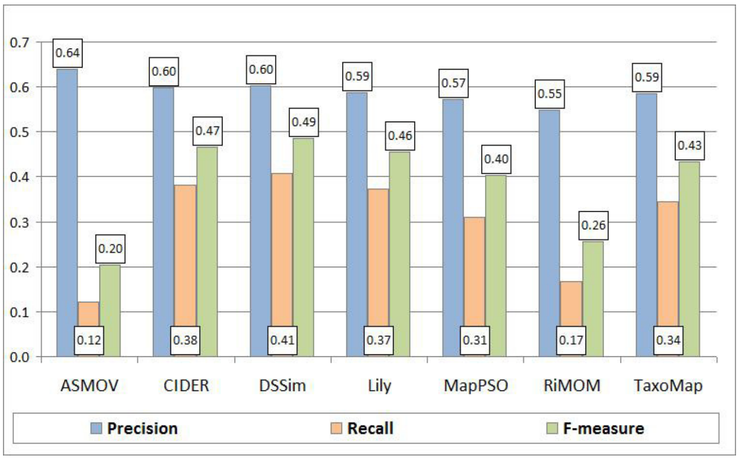


Fig. 3. Matching quality results.

We can observe from Table 9, that all the systems that participated in the directory track in 2007 and 2008 (ASMOV, DSSim, Lily and RiMOM), have increased their precision values. Considering recall, we can see that in general the systems that had participated in 2007 and 2008 directory tracks, have decreased their values, the only system that increased its recall values is DSSim. In fact, DSSim is the system with the highest F-measure value in 2008.

Table 9 shows that in total 21 matching systems have participated during the 4 years (2005 - 2008) of the OAEI campaign in the directory track. No single system has participated in all campaigns involving the web directory dataset (2005 - 2008). A total of 14 systems have participated only one time in the evaluation, 5 systems have participated 2 times, and only 2 systems have participated 3 times. The systems that have participated in 3 evaluations are Falcon (2005, 2006 and 2007) and RiMoM (2006, 2007, 2008), the former with a constant increase in the quality of the results, the later with a constant increase in precision, but in the last evaluation (2008) recall dropped significantly from 71% in 2007, to 17% in 2008.

System	**Recall**				**Precision**			**F-Measure**		
Year →	2005	2006	2007	2008	2006	2007	2008	2006	2007	2008
ASMOV			0.44	0.12		0.59	0.64		0.50	0.20
automs		0.15			0.31			0.20		
CIDER				0.38			0.60			0.47
CMS	0.14									
COMA		0.27			0.31			0.29		
ctxMatch2	0.09									
DSSim			0.31	0.41		0.60	0.60		0.41	0.49
Dublin20	0.27									
Falcon	0.31	0.45	0.61		0.41	0.55		0.43	0.58	
FOAM	0.12									
hmatch		0.13			0.32			0.19		
Lily			0.54	0.37		0.57	0.59		0.55	0.46
MapPSO				0.31			0.57			0.40
OCM		0.16			0.33			0.21		
OLA	0.32		0.84			0.62			0.71	
OMAP	0.31									
OntoDNA			0.03			0.55			0.05	
Prior		0.24	0.71		0.34	0.56		0.28	0.63	
RiMOM		0.40	0.71	0.17	0.39	0.44	0.55	0.40	0.55	0.26
TaxoMap				0.34			0.59			0.43
X-SOM			0.29			0.62			0.39	
Average	*0.22*	*0.26*	*0.50*	*0.30*	*0.35*	*0.57*	*0.59*	*0.29*	*0.49*	*0.39*
#	*7*	*7*	*9*	*7*	*7*	*9*	*7*	*7*	*9*	*7*

Table 9. Summary of submissions by year (no precision was computed in 2005). The Prior line covers Prior+ as well and the OLA line covers OLA_2 as well.

As can be seen in Figure 4 and Table 9, there is an increase in the average precision for the directory track of 2008, along with a decrease in the average recall compared to 2007. Notice that in 2005 the data set allowed only the estimation of recall, therefore Figure 4 and Table 9 do not contain values of precision and F-measure for 2005.

A comparison of the results in 2006, 2007 and 2008 for the top-3 systems of each year based on the highest values of the F-measure indicator is shown in Figure 5. The key observation here is that unfortunately the top-3 systems of 2007 did not participate in the directory task this year, therefore, the top-3 systems for 2008 is a new set of systems (Lily, CIDER and DSSim). From these 3 systems, CIDER is a newcomer, but Lily and DSSim had also participated in the directory track of 2007, when they did not manage to enter into the top-3 list.

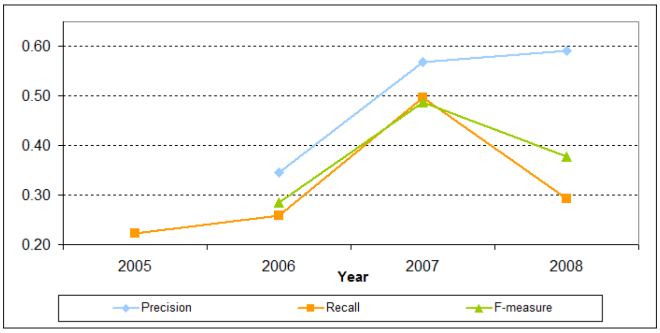


Fig. 4. Average results of the top-3 systems per year.

The quality of the best F-measure result of 2008 (0.49) demonstrated by DSSim is lower than the best F-measure of 2007 (0.71) by OLA$_2$ but still higher than that of 2006 by Falcon (0.43). The best precision result of 2008 (0.64) demonstrated by ASMOV is higher than the results obtained in 2007 (0.62) by both OLA$_2$ and X-SOM. Finally, for what concerns recall, the best result of 2008 (0.41) demonstrated by DSSim is also lower than the best results obtained in 2007 (0.84) obtained by OLA$_2$ and in 2006 (0.45) by Falcon. This decrease in the maximum values achieved by the participating systems may be caused by participants tuning their system parameters for more diverse tasks this year. Hence, the overall results of systems could have improved at the expense of results in the directory track. For example, we can observe that both ASMOV and Lily have very good results (over 90% F-measure) for the Benchmark-2008 track, which are higher than the Benchmarck-2007 track.

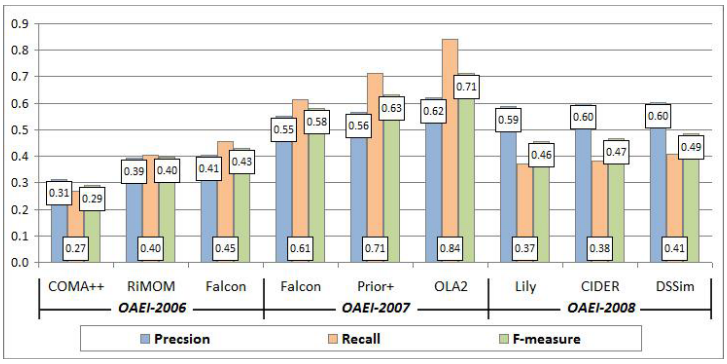


Fig. 5. Comparison of matching quality results in 2006, 2007 and 2008.

Partitions of positive and negative correspondences according to the system results are presented in Figure 6 and Figure 7, respectively.

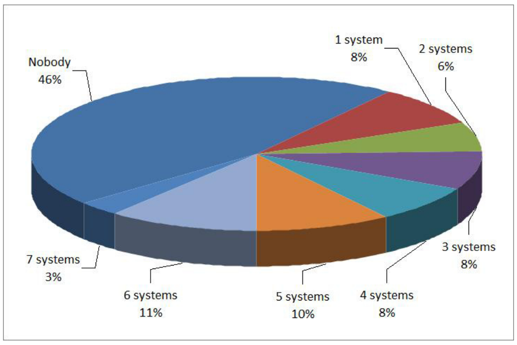


Fig. 6. Partition of the system results on positive correspondences.

Figure 6 shows that the systems managed to discover only 54% of the total number of positive correspondences (Nobody = 46%). Only 11% of positive correspondences were found by almost all (6) matching systems, while 3% of the correspondences were found by all the participants in 2008. This high percentage of positive correspondences not found by the systems correspond to the low recall values we observe in Table 9, which are the cause of the decrease in average recall from 2007 to 2008. Figure 7 shows that most of the negatives correspondences were not found by the systems (correctly). Figure 7 also shows that six systems found 11% of negative correspondences, i.e., mistakenly returned them as positive. The last two observations suggest that the discrimination ability of the dataset remains still high as in previous years.

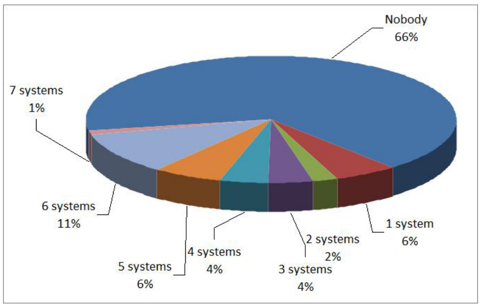


Fig. 7. Partition of the system results on negative correspondences.

Let us now compare partitions of the system results in 2006, 2007 and 2008 on positive and negative correspondences, see Figure 8 and Figure 9, respectively.

Figure 8 shows that 46% of positive correspondences have not been found by any of the matching systems in 2006, while in 2007 all the positive correspondences have been collectively found. In 2008, 46% of the positive correspondences have not been found by the participating systems, as in 2006. This year, systems performed in the line of 2006. In 2007, the results were exceptional because the participating systems alltogether had a full coverage of the expected results and very high precision and recall. Unfortunately, the best systems of last year did not participate this year and the other systems do not seem to cope with the previous results.

Figure9 shows that in 2006 in overall the systems have correctly not returned 26% of negative correspondences, while in 2007, this indicator decreased to 2%; in turn in 2008 the value increased to 66%, this is, the set of participating systems in 2008 cor-

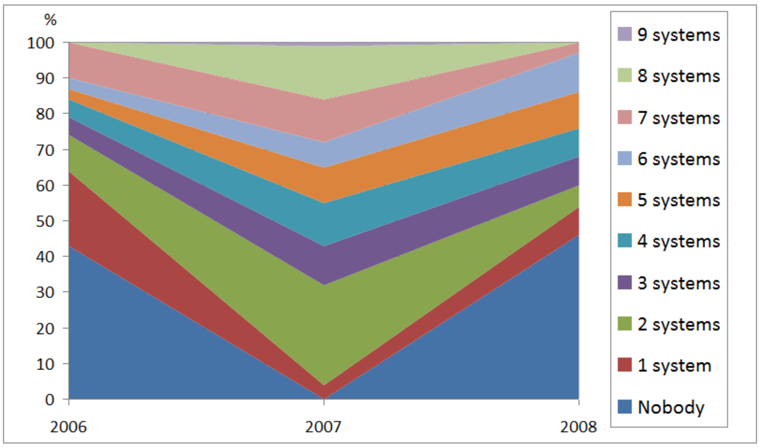


Fig. 8. Comparison of partitions of the system results on positive correspondences in 2006, 2007 and 2008.

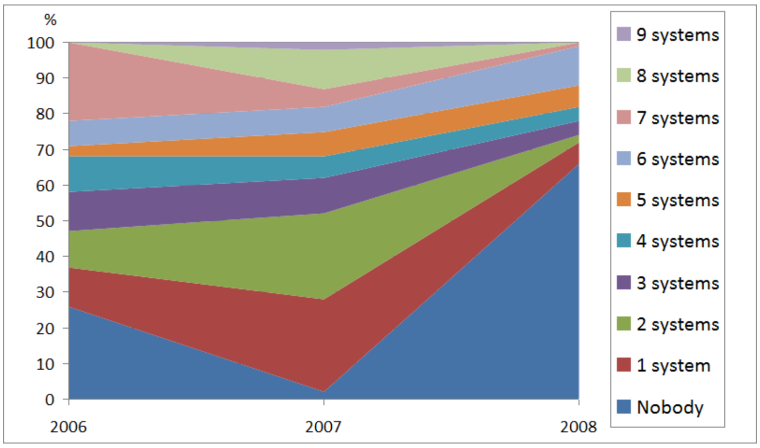


Fig. 9. Comparison of partitions of the system results on negative correspondences in 2006, 2007 and 2008.

rectly avoid more negative correspondences than those participating in 2006 and 2007. In 2006, 22% of negative correspondences were mistakenly found by all (7) the matching systems, while in 2007, this indicator decreased to 5% (for 7 systems), and in 2008, the value decreased even more to 1%. An interpretation of these observations could be that the set of participating systems in 2008 have a more "cautious" strategy than in 2007 and 2006. In 2007 we can observe that the set systems showed a more "brave" strategy in discovering correspondences, were the set of positive correspondences was fully covered, but covering mistakenly also 98% of the negative correspondences, while in 2008 the set of participating systems covered just 54% of the positive correspondences, but covering only 34% of negative correspondences.

6.3 Comments

An important observation from this evaluation is that ontology matching is still making progress on the web directory track this year, if we consider that the set of participating systems in 2008 is almost completely different compared to 2007. With respect to the average performance of the systems (given by F-Measure in Figure 4), the set of participating systems in 2008 performed worse than the set of participating systems in 2007, but better than those participating in 2006. This suggests that the systems participating in 2008 experienced a higher number of difficulties on the test case, in comparison to 2007, which means that there is still room for further improvements, specially in recall. A considerable remark this year is that it is hard for a single system to perform well in all the situations when finding correspondences is needed (which are simulated by the different OAEI tracks); this suggests that a general purpose matching system is difficult to construct. Finally, as partitions of positive and negative correspondences indicate (see Figure 6 and Figure 7), the dataset still retains a good discrimination ability, i.e., different sets of correspondences are still hard for the different systems.

7 Multilingual directories

The multilingual directory data set (mldirectory) is a data set created from real internet directory data. This data provides alignment problems for different internet directories. This track mainly fpcuses on multilingual data (English and Japanese) and instances.

7.1 Test data and experimental settings

The multilingual directory data set is constructed from Google (open directory project), Yahoo!, Lycos Japan, and Yahoo! Japan. The data set consists of five domains: automobile, movie, outdoor, photo and software, which are used in [11, 10]. There are four files for each domain. Two are for English directories and the rest are for Japanese directories. Each file is written in OWL. A file is organized into two parts. The first part describes the class structures, which are organized with `rdfs:subClassOf` relationships. Each class might also have `rdfs:seeAlso` properties, which indicate related classes. The second part is the description of instances of the classes. Each description has an instance ID, class name, instance label, and short description.

There are two main differences between the *mldirectory* data set and *directory* data set, which is also available for OAEI-2008.

- The first one is a multilingual set of directory data. As we mentioned above, the data set has four different ontologies with two different languages for one domain. As a result, we have six alignment problems for one domain. These include one English-English alignment, four English-Japanese alignments, and one Japanese-Japanese alignment.
- The second difference is the instances of classes. In the multilingual directory data set, the data not only has relationships between classes but also instances in the classes. As a result, we can use snippets of web pages in the Internet directories as well as category names in the multilingual directory data set.

We encouraged participants to submit alignments for all domains. Since there are five domains and each domain has six alignment patterns, this is thirty alignments in total. However, participants can submit some of them, such as the English-English alignment only.

Participants are allowed to use background knowledge such as Japanese-English dictionaries and WordNet. In addition, participants can use different data included in the multilingual directory data set for parameter tuning. For example, the participants can use automobile data for adjusting the participant's system, and then induce the alignment results for movie data by the system. Participants cannot use the same data to adjust their system, because the system will consequently not be applicable to unseen data. In the same manner, participants cannot use specifically crafted background knowledge because it will violate the assumption that we have no advanced knowledge of the unseen data.

7.2 Results

In the 2008 campaign, four participants dealt with the mldirectory data set: DSSim, Lily, MapPSO and RiMOM. Among the four systems, three of them – DSSim, MapPSO, and RiMOM – were used for all five domains in the English-English alignment, and one of them, Lily, was used in the task for two domains, automobile and movie. The number of correspondences found by the systems are shown in Table 10. As can be seen in this table, Lily finds more correspondences than do the other systems. Conversely, MapPSO retrieves only a few correspondences from the data set.

In order to learn the different biases of the systems, we counted the number of common correspondences retrieved by the systems. The results are shown in Table 11. The letters D, L, M and R in the top row denote system names DSSim, Lily, MapPSO, and RiMOM, respectively. For example, the DR column is the number of correspondences retrieved by both DSSim and RiMOM. We can see that both systems retrieve the same 82 correspondences in the movie domain. In this table, we see interesting phenomena. Lily and RiMOM have the same bias. For example, in the auto domain, 33% of the correspondences found by Lily were also retrieved by RiMOM, and 46% of the correspondences found by RiMOM were also retrieved by Lily. The same phenomenon is

	DSSim	Lily	MapPSO	RiMOM
Auto	188	377	265	275
Movie	1181	1864	183	1681
Outdoor	268	-	10	538
Photo	141	-	38	166
Software	372	-	60	536
Total	2150	2241	556	3196

Table 10. Number of correspondences found (English-English alignments).

also seen in the movie domain. In contrast, MapPSO has a very different tendency. Although the system found 556 alignments in total, only one correspondence was found by the other systems.

	D	L	M	R	DL	DM	DR	LM	LR	MR	DLM	DLR	DMR	LMR	DLMR
Auto	139	208	264	104	5	0	7	0	126	0	0	37	1	0	0
Movie	946	988	183	734	11	0	82	0	723	0	0	142	0	0	0
Outdoor	260	0	10	530	0	0	8	0	0	0	0	0	0	0	0
Photo	137	0	38	162	0	0	4	0	0	0	0	0	0	0	0
Software	338	0	60	502	0	0	34	0	0	0	0	0	0	0	0

Table 11. Number of common correspondences retrieved by the systems. D, L, M, and R denote DSSim, Lily, MapPSO, and RiMOM, respectively.

We also created a component bar chart (Figure 10) for clarifying the sharing of retrieved correspondences. In the automobile and movie domains, 80% of the correspondences are found by only one system, and most of the other 20% are found by both Lily and RiMOM. From this graph, we can see that Lily has the same bias as RiMOM, but the system still found many correspondences that the other systems did not find. For the remaining domains, outdoor, photo and software, the correspondences found by only one system reached almost 100%.

Unfortunately, the results of other alignment tasks such as English-Japanese alignments (ontology 1-3, ontology 1-4, ontology 2-3, and ontology 2-4), Japanese-Japanese alignments (ontology 3-4) were only submitted by RiMOM. The number of alignments by RiMOM are shown in Table 12.

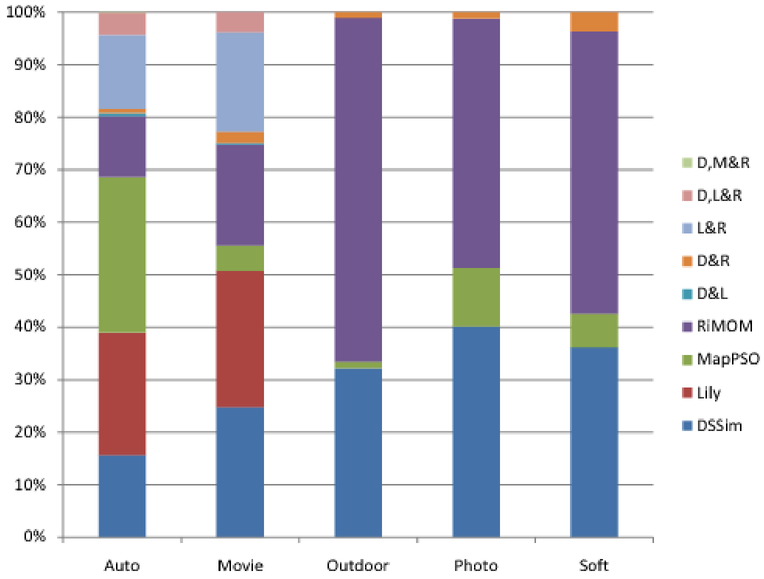


Fig. 10. Shared correspondences.

Domain	ont 1-2	ont 1-3	ont 1-4	ont 2-3	ont 2-4	ont 3-4	Total
Auto	275	99	242	79	225	262	1182
Movie	1681	35	30	35	59	65	1905
Outdoor	538	25	64	25	97	31	780
Photo	166	15	17	15	31	20	264
Software	536	104	125	78	100	84	1027

Table 12. Number of alignments by RiMOM.

8 Library

8.1 Data set

This test case deals with two large Dutch thesauri. The National Library of the Netherlands (KB) maintains two large collections of books: the Scientific Collection and the Deposit collection, containing respectively 1.4 and 1 million books. Each collection is annotated – *indexed* – using its own controlled vocabulary. The former is described using the GTT thesaurus, a huge vocabulary containing 35,194 general concepts, ranging from "Wolkenkrabbers" (Sky-scrapers) to "Verzorging" (Care). The latter is indexed against the Brinkman thesaurus, which contains a large set of headings (5,221) for describing the overall subjects of books. Both thesauri have similar coverage (2,895 concepts actually have exactly the same label) but differ in granularity.

Each concept has exactly one preferred label, plus synonyms, extra hidden labels or scope notes. The language of both thesauri is Dutch,[10] which makes this track ideal for testing alignment in a non-English situation. Concepts are also provided with structural information, in the form of *broader* and *related* links. However, GTT (resp. Brinkman) contains only 15,746 (resp 4,572) hierarchical *broader* links and 6,980 (resp. 1,855) associative *related* links. The thesauri's structural information is thus very poor.

For the purpose of the OAEI campaign, the two thesauri were made available in SKOS format. OWL versions were also provided, according to the – lossy – conversion rules detailed on the web site[11].

In addition, we have provided participants with *book descriptions*. At KB, almost 250000 books belong both to KB Scientific and Deposit collections, and are therefore already indexed against both GTT and Brinkman. Last year, we have used these books as a reference for evaluation. However, these books can also be a precious hint for obtaining correspondences. Indeed one of last year's participant had exploited co-occurrence of concepts, though on a collection obtained from another library. This year, we split the 250000 books in two sets: two third of them are provided to participants for alignment computation, and one third is kept as a test set to be used as a reference for evaluation.

8.2 Evaluation and results

Three systems provided final results: DSSim (2,930 `exactMatch` correspondences), Lily (2,797 `exactMatch` correspondences) and TaxoMap (1,872 `exactMatch` correspondences, 274 `broadMatch`, 1,031 `narrowMatch` and 40 `relatedMatch` correspondences).

We have followed the scenario-oriented approach followed for 2007 library track, as explained in [12].

Evaluation in a thesaurus merging scenario. The first scenario is *thesaurus merging*, where an alignment is used to build a new, unified thesaurus from GTT and Brinkman

[10] A quite substantial part of GTT concepts (around 60%) also have English labels.

[11] `http://oaei.ontologymatching.org/2008/skos2owl.html`

thesauri. Evaluation in such a context requires assessing the validity of each individual correspondence, as in "standard" alignment evaluation.

As last year, there was no reference alignment available. We opted for evaluating precision using a reference alignment based on a lexical procedure. This makes use of direct comparison between labels, but also exploits a Dutch morphology database that allows to recognize variants of a word, e.g., singular and plural. 3.659 reliable equivalence links are obtained this way. We also measured coverage, which we define as the proportion of all good correspondences found by an alignment divided by the total number of good correspondences produced by all participants and those in the reference – this is similar to the pooling approach that is used in major Information Retrieval evaluations, like TREC.

For manual evaluation, the set of all *equivalence* correspondences[12] was partitioned into parts unique to each combination of participant alignments, and each part was sampled. A total of 403 correspondences were assessed by one Dutch native expert.

From these assessments, precision and pooled recall were calculated with their 95% confidence intervals, taking into account sampling size. The results are shown in Table 13, which identifies DSSim as performing better than both other participants.

Alignment	Precision			Pooled recall		
DSSim	93.3%	±	0.3%	68.0%	±	1.6%
Lily	52.9%	±	3.0%	36.8%	±	2.2%
TaxoMap (exactMatch)	88.1%	±	0.8%	41.1%	±	1.0%

Table 13. Precision and coverage for the thesaurus merging scenario.

DSSim has performed better than last year. This result stems probably from DSSim now proposing almost only exact lexical matches of SKOS labels, as opposed to last year.

For the sake of completeness, we also evaluated the precision of the TaxoMap correspondences that are not of type `exactMatch`. We categorized them according to the strength that TaxoMap gave them (0.5 or 1). 20% (±11%) of the correspondences with strength 1 are correct. The figure rises to 25.1% (±8.3%) when considering all non-`exactMatch` correspondences, which hints at the strength not being very informative.

Evaluation in an annotation translation scenario. The second usage scenario is based on an *annotation translation* process supporting the re-indexing of GTT-indexed books with Brinkman concepts [12].

This evaluation scenario interprets the correspondences provided by the different participants as rules to translate existing GTT book annotations into equivalent Brinkman annotations. Based on the quality of the results for books we know the correct annotations of, we can assess the quality of the initial correspondences.

[12] We did not proceed with manual evaluation of the *broader*, *narrower* and *related* links at once, as only one contestant provided such links.

Evaluation settings and measures. The simple concept-to-concept correspondences sent by participants were transformed into more complex mapping rules that associate one GTT concept and a set of Brinkman concepts – some GTT concepts are indeed involved in several mapping statements. Considering `exactMatch` only, this gives 2,930 rules for DSSim, 2,797 rules for Lily and 1,851 rules for TaxoMap. In addition, TaxoMap produces resp. 229, 897 and 39 rules considering `broadMatch`, `narrowMatch` and `relatedMatch`.

The set of GTT concepts attached to each book is then used to decide whether these rules are *fired* for this book. If the GTT concept of one rule is contained by the GTT annotation of a book, then the rule is fired. As several rules can be fired for a same book, the union of the consequents of these rules forms the translated Brinkman annotation of the book.

On a set of books selected for evaluation, the generated concepts for a book are then compared to the ones that are deemed as correct for this book. At the book level, we measure how many books have a rule fired on them, and how many of them are actually *matched* books, i.e., books for which the generated Brinkman annotation contains at least one correct concept. These two figures give a precision (P_b) and a recall (R_b) for this book level.

At the annotation level, we measure (i) how many translated concepts are correct over the annotation produced for the books on which rules were fired (P_a), (ii) how many correct Brinkman annotation concepts are found for all books in the evaluation set (R_a), and (iii) a combination of these two, namely a Jaccard overlap measure between the produced annotation (possibly empty) and the correct one (J_a).

The ultimate measure for alignment quality here is at the annotation level. Measures at the book level are used as a raw indicator of users' (dis)satisfaction with the built system. A R_b of 60% means that the alignment does not produce any useful candidate concept for 40% of the books. We would like to mention that, in these formulas, results are counted on a book and annotation basis, and not on a rule basis. This reflects the importance of different thesaurus concepts: a translation rule for a frequently used concept is more important than a rule for a rarely used concept. This option suits the application context better.

Manual evaluation. Last year, we evaluated the results of the participants in two ways, one manual – KB indexers evaluating the generated indices – and one automatic – using books indexed against both GTT and Brinkman. This year, we have not performed manual investigation. Findings of last year can be found in [12].

Automatic evaluation and results. Here, the reference set consists of 81,632 dually-indexed books forming the test set presented in Section 8.1. The existing Brinkman indices from these books are taken as a reference to which the results of annotation translation are automatically compared.

The upper part of Table 14 gives an overview of the evaluation results when we only use the `exactMatch` correspondences. DSSim and TaxoMap perform similarly in precision, and much ahead of Lily. If precision almost reaches last year's best results, recall is much lower. Less than one third of the books were given at least one correct Brinkman concept in the DSSim case. At the annotation level, half of the translated concepts are not validated, and more than 75% of the real Brinkman annotation is not found. We already pointed out that the correspondences from DSSim are mostly generated by lexical similarity. This indicates, as last year, that lexically equivalent correspondences alone do not solve the annotation translation problem.

Participant	P_b	R_b	P_a	R_a	J_a
DSSim	56.55%	31.55%	48.73%	22.46%	19.98%
Lily	43.52%	15.55%	39.66%	10.71%	9.97%
TaxoMap	52.62%	19.78%	47.36%	13.83%	12.73%
TaxoMap+broadMatch	46.68%	19.81%	40.90%	13.84%	12.52%
TaxoMap+hierarchical	45.57%	20.23%	39.51%	14.12%	12.67%
TaxoMap+all correspondences	45.51%	20.24%	39.45%	14.13%	12.67%

Table 14. Results of annotation translations generated from correspondences.

Among the three participants, only TaxoMap generated `broadMatch` and `narrowMatch` correspondences. To evaluate their usefulness for annotation translation, we evaluated their influence when they were added to a common set of rules. As shown in the four *TaxoMap* lines in Table 14, the use of `broadMatch`, `narrowMatch` and `relatedMatch` correspondences slightly increases the chances of having a book given a correct annotation. However, this unsurprisingly results in a loss of precision.

8.3 Discussion

The first comment on this track concerns the *form* of the alignment returned by the participants, especially with respect to the type and cardinality of alignments. All three participants proposed alignments using the SKOS links we asked for. However, only one participants proposed hierarchical `broader`, `narrower` and `related` links. Experiments show that these links can be useful for the application scenarios at hand. The `broader` links are useful to attach concepts which cannot be mapped to an equivalent corresponding concept but a more general or specific one. This is likely to happen, since the two thesauri have different granularity but a same general scope.

This actually mirrors what happened in last year's campaign, where only one participant had given non-exact correspondence links – even though it was `relatedMatch` then. Evaluation had shown that even though the general quality was lowered by considering them, the loss of precision was not too important, which could make these links interesting for some application variants, *e.g.* semi-automatic re-indexing.

Second, there is no precise handling of one-to-many or many-to-many alignments, as last year. Sometimes a concept from one thesaurus is mapped to several concepts from the other. This proves to be very useful, especially in the annotation translation scenario where concepts attached to a book should ideally be translated as a whole.

Finally, one shall notice the low coverage of alignments with respect to the thesauri, especially GTT: in the best case, only 2,930 of its 35K concepts were linked to some Brinkman concept, which is less than last year (9,500). This track, arguably because of its Dutch language context, is difficult. We had hoped that the release of a part of the set of KB's dually indexed books would help tackle this difficulty, as previous year's campaign had shown promising results when exploiting real book annotations. Unfortunately none of this year's participants have used this resource.

9 Very large crosslingual resources

The goal of the Very Large Crosslingual Resources task is twofold. First, we are interested in the alignment of vocabularies in different languages. Many collections throughout Europe are indexed with vocabularies in languages other than English. These collections would benefit from an alignment to resources in other languages to broaden the user group, and possibly enable integrated access to the different collections.

Second, we intend to present a realistic use case in the sense that the resources are large, rich in semantics but weak in formal structure, i.e., realistic on the Web. For collections indexed with an in-house vocabulary, the link to a widely-used and rich resource can enhance the structure and increase the scope of the in-house thesaurus.

9.1 Data set

Three resources are used in this task:

GTAA The GTAA is a Dutch thesaurus used by the Netherlands Institute for Sound and Vision to index their collection of TV programs. It is a facetted thesaurus, of which we use the following four themes: (1) **Subject**: the topic of a TV program, $\approx$ 3800 terms; (2) **People**: the main people mentioned in a TV program, $\approx$ 97.000 terms; **Names**: the main "Named Entities" mentioned in a TV program (Corporation names, music bands, etc.), $\approx$ 27.000 terms; **Location**: the main locations mentioned in a TV program or the place where it has been created, $\approx$ 14.000 terms.

WordNet WordNet is a lexical database of the English language developed at Princeton University[13]. Its main building blocks are synsets: groups of words with a synonymous meaning. In this task, the goal is to match noun-synsets. WordNet contains 7 types of relations between noun-synsets, but the main hierarchy in WordNet is built on hyponym relations, which are similar to subclass relations. W3C has translated WordNet version 2.0 into RDF/OWL[14].

The original WordNet model is a rich and well-designed model. However, some tools may have problems with the fact that the synsets are instances rather than classes. Therefore, for the purpose of this OAEI task, we have translated the hyponym hierarchy in a `skos:broader` hierarchy, making the synsets `skos:Concepts`.

DBpedia DBPedia contains 2.18 million resources or "things", each tied to an article in the English language Wikipedia. The "things" are described by titles and abstracts in English and often also in other languages, including Dutch. DBPedia "things" have numerous properties, such as categories, properties derived from the wikipedia 'infoboxes', links between pages within and outside wikipedia, etc. The purpose of this task is to map the DBPedia "things" to WordNet synsets and GTAA concepts.

[13] http://wordnet.princeton.edu/

[14] http://www.w3.org/2006/03/wn/wn20/

9.2 Evaluation Setup

We evaluate the results of the three alignments (GTAA-WordNet, GTAA-DBPedia, WordNet-DBPedia) in terms of precision and recall. We present measures for each GTAA facet separately, instead of a global value, because each facet could lead to very different performance.

In the precision and recall calculations, we use a kind of semantic distance; we take into account the distance between a correspondence that we find in the results and the ideal correspondence that we would expect for a certain concept. For each equivalence relation between two concepts in the results, we determine if (i) one is equivalent to the other, (ii) one is a broader/narrower concept than the other, (iii) one is in none of the above ways related to the other. In case (i) the correspondence counts as 1, in case (ii) the correspondence counts as 0.5 and in case (iii) as 0.

Precision We take samples of 100 correspondences per GTAA facet for both the GTAA-DBPedia and the GTAA-WordNet alignments and evaluate their correctness in terms of exact match, broader, narrower or related match, or no match. The alignment between WordNet and DBPedia is evaluated by inspection of a random sample of 100 correspondences.

Recall Due to time constraints, we only determine recall of two of the four GTAA facets: People and Subjects. These are the most extreme cases in terms of size and precision values. We create a small reference alignment from a random sample of 100 GTAA concepts per facet, which we manually map to WordNet and DBPedia. The result of the GTAA-WordNet and GTAA-DBPedia alignments are compared to the reference alignments. We do not provide a recall measure for the DBPedia-WordNet correspondence.

9.3 Results

Only one participant, DSSim, participated in the VLCR task. The evaluation of the results therefore focuses on the differences between the three alignments, and the four facets of the GTAA. Table 15 shows the number of concepts in each resource and the number of correspondences returned for each resource pair. The largest number of correspondences was found between DBpedia and WordNet (28,974), followed by GTAA-DBPedia (13,156) and finally GTAA-WordNet (2,405). We hypothesize that the low number of the latter pair is due to the multilingual nature. Except for 9 concepts, all GTAA concepts that were mapped to DBPedia were also mapped to WordNet.

Precision The precision of the GTAA-DBPedia alignment is higher than that of the GTAA-WordNet alignment. A possible explanation is the high number of disambiguation errors for WordNet, which is much finer grained than for GTAA or DBPedia.

A remarkable difference can be seen in the People facet. It is the worst scoring facet in the GTAA-WordNet alignment (10%), while it is the best facet in GTAA-DBPedia (94%). Inspection of the results revealed what caused the many mistakes for WordNet: almost none of the people in GTAA are present in WordNet. Instead of giving up, DSSim continues to look for a correspondence and maps the GTAA person to a lexically similar word in WordNet. This problem is apparently not present in DBPedia. Although we do not yet fully understand why not, an important factor is that more Dutch people are represented in DBPedia.

Vocabulary		#concepts	#corr to WN	#corr to DBP	#corr to GTAA
Wordnet		82.000	n.a.	28974	2405
DBPedia		2180.000	28974	n.a.	13156
GTAA		160.000	2405	13156	n.a.
Facet:	Subject	3800	655	1363	n.a.
	Person	97.000	82	2238	n.a.
	Name	27.000	681	3989	n.a.
	Location	14.000	987	5566	n.a.

Table 15. Number of correspondences in each alignment.

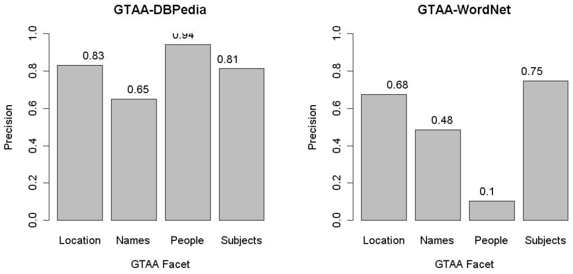


Fig. 11. Estimated precision of the alignment between GTAA and DBpedia (left) and WordNet (right).

Apart from the People facet, the differences between the facets are consistent over the GTAA-DBPedia and GTAA-WordNet alignments. Subjects and Locations score high, Names somewhat less.

The alignment between DBPedia and WordNet had a precision of 45%. DBPedia contains type links (wordnet-type and `rdf:type`) to WordNet synsets. There was no overlap between the alignment submitted by DSSim and these existing links.

Recall We created reference alignments by matching samples of 100 concepts from the People and Subjects facets to both DBPedia and WordNet. However, none of the People in our sample of 100 GTAA People could be mapped to WordNet. Therefore, recall for this particular alignment could not be detemined.

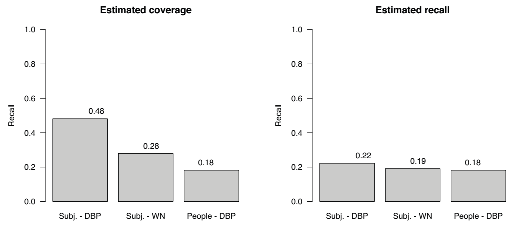


Fig. 12. Estimated coverage (left) and recall (right) for the alignments between the Subject facet of GTAA and DBpedia and WordNet, and for the alignment between the People facet of GTAA and DBpedia.

Figure 12 shows how many of the GTAA Subject and People in our reference alignment were also found by DSSim. We call this *coverage*. The second figure depicts how many GTAA concept in our reference alignment were found by DSSim to the exact same DBPedia/WordNet concept, which is the conventional definition of recall. All three alignments had a similar recall score of aroud 20%.

9.4 Summary of the results

Tables 16 and 17 summarize the result.

	Precision			
Alignment	Subjects	People	Location	Names
GTAA-DBPedia	0.81 (11.6%)	0.94 (7.02%)	0.83 (11.1%)	0.65 (14.1%)
GTAA-WordNet	0.75 (12.8%)	0.1 (8.8%)	0.68 (13.8%)	0.48 (14.7%)

Table 16. Summary of the participant's precision scores (numbers in parentheses represent the different error margins).

	Recall		Estimated coverage	
Alignment	Subjects	People	Subjects	People
GTAA-DBPedia	0.22 (12.2%)	0.18 (11.3%)	0.48 (14.7%)	0.18 (11.3%)
GTAA-WordNet	0.19 (11.6%)	NA	0.28 (13.2%)	NA

Table 17. Summary of the participant's estimated recall and coverage scores (numbers in parentheses represent the different error margins).

9.5 Discussion

Other types of correspondence relations The VLCR task once more confirmed what was already known: more correspondence types are necessary than only exact matches. While inspecting alignments, we found many cases where a link between two concepts seems useful for a number of applications, without being equivalent. For example:

```
Subject:pausbezoeken[15]
  and List_of_pastoral_visits_of_Pope_John_Paul_II_outside_Italy.
Location:Venezuela and synset-Venezuelan-noun-1
Subject:Verdedigingswerken[16] and fortification
```

Using context When looking at the types of mistakes that were made, it became clear that a number of them could have been avoided by using the specific structure of the resources being matched. The fact that the GTAA is organized in facets, for example, can be used to disambiguate terms that appear both as a person and as a location. This information is represented by the `skos:inScheme` property. Examples of incorrect correspondences that might have been avoided if facet information was used are:

```
Person:GoghVincentvan -> synset-vacationing-noun-1
Location:Harlem -> synset-hammer-noun-8
Location:Melbourne -> synset-Melbourne-noun-1[17]
```

Another example of resource-specific structure that could help matching are the redirects between pages in Wikipedia or between "things" in DBPedia. DBPedia contains things for which no other information is available than a 'redirect' property pointing to another thing. The wikipedia page for "Gordon Summer" for example, is immediately referred to the page for "Sting, the musician". The titles of these referring pages could well serve as alternative labels, and thus aid the correspondence between the gtaa concept person:SummerGordon and the dbepdia thing Sting(musician).

Of course, there is a trade-off between the amount of resource-specific features that are taken into account and the general applicability of the matcher. However, some of the features discussed above, such as facet information, are found in a wide range of thesauri and are therefore serious candidates for inclusion in a tool.

Reflection on the evaluation Deciding which synset or DBpedia thing is the most suitable match for a GTAA concept is a non-trivial task, even for a human evaluator.

[15] Pope visits, in English.

[16] Defenses, in English.

[17] This synset indeed refers to "a resort town in east central Florida".

Often, multiple correspondences are reasonable. Therefore, the recall figures that are based on a hand-made reference alignment give a possibly too negative impression of the quality of the alignment. The evaluation task was further complicated because of the 'related' matches. There is a lack of clear definitions of when two concepts are related.

Another factor that has to be considered when interpreting the precision and recall figures, is the number of Dutch-specific concepts in the GTAA. For example, the concept Name:Diogenes denotes a Dutch TV program instead of the ancient Greek. Although the fact that Diogenes is in the Name facet and not in the People facet provides a clue of its intended meaning, it could be argued that this type of Dutch-specific concepts pose an unfair challenge to matchers.

During the evaluation process, we found cases in which DSSim mapped to a DBPedia disambiguation page instead of an actual article. We consider this to be incorrect, since it leaves the disambiguation task to the user.

10 Conference

The conference track involves matching several ontologies from the conference organization domain. Participant results have been evaluated along different modalities and a consensus workshop aiming at studying the elaboration of consensus when establishing reference alignments has been organised.

10.1 Test set

The collection consists of fifteen ontologies in the domain of organizing conferences. Ontologies have been developed within the OntoFarm project[18]. In contrast to last year's conference track, there is one new ontology and several new methods of evaluation.

The main features of this data 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 alignments 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 terminologies.
- *Relative richness in axioms.* Most ontologies were equipped with description logic axioms of various kinds, which opens a way to use semantic matchers.

Ontologies differ in number of classes, of properties, in their expressivity, but also in underlying resources. Ten 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 three are based on *web pages* of concrete conferences.

Participants had to provide either complete alignments or interesting correspondences (nuggets), for all or some pairs of ontologies. Participants could also take part in two different tasks. First, participants could find correspondences without any specific

[18] `http://nb.vse.cz/~svatek/ontofarm.html`

application context given (generic correspondences). Second, participants could find out correspondences with regard to an application scenario: *transformation application*. This means that final correspondences are to be used for conference data transformation from one software tool for organizing conference to another one.

This year, results of participants were evaluated by five different methods: evaluation based on manual labeling, reference alignments, data mining method, logical reasoning, and on consensus of experts.

10.2 Evaluation and results

We had three participants. All of them delivered generic correspondences. Aside from results from evaluation methods (sections below) we deliver some simple observations about participants:

- DSSim and Lily delivered in total 105 alignments. All ontologies were matched to each other. ASMOV delivered 75 alignments. For our evaluation we do not consider alignments in which ontologies were matched to themselves.
- Two participants delivered correspondences with certainty factors between 0 and 1 (ASMOV and Lily); one (DSSim) delivered correspondences with confidence measures 0 or 1, where 0 is used to describe a correspondence as negative.
- DSSim and Lily delivered only equivalence, e.g., no subsumption, relations, while ASMOV also provided subsumption relations[19].
- All participants delivered class-to-class correspondences and property-to-property correspondences.

Evaluation based on manual labeling This kind of evaluation is based on sampling and manual labeling of random samples of correspondences because the number of all distinct correspondences is quite high. Particularly, we followed the method of *Stratified random sampling* described in [20]. Correspondences of each participant were divided into three subpopulations (strata) according to confidence measures[20]. For each stratum we randomly chose 75 correspondences in order to have 225 correspondences for manual labeling for each system; except the one stratum of the DSSim system with 150 correspondences.

In Table 18 there are data for each stratum and system where *Nh* is the size of the stratum, *nh* is the number of sample correspondences from the stratum, *TP* is the number of correct correspondences from sample from the stratum, and *Ph* is an approximation of precision for the correspondences in the stratum. Furthermore, based on the assumption that this adheres to *binomial distribution* we computed *margin of errors* (with confidence of 95%) for the approximated precision for each system based on equations from [20]. In Table 19 there are measures for the entire populations. We computed approximated precision $P*$ in the entire population as weighted average from the approximated precisions of each strata. Finally, we also computed so-called 'relative'

[19] Finally, no current evaluation methods did take into account subsumption correspondences. Considering these correspondences in evaluation methods is our plan for next year of the conference track.

[20] DSSim provided merely 'certain' correspondences, so there is just one stratum for this system.

	(0,0.3]		(0.3,0.6]		(0.6,1.0]		
system	ASMOV	Lily	ASMOV	Lily	ASMOV	Lily	DSSim
Nh	779	426	349	911	135	407	1950
nh	75	75	75	75	75	75	150
TP	16	33	38	27	51	39	46
Ph	21%	44%	51%	36%	68%	52%	30%
	±12%	±12%	±12%	±12%	±12%	±12%	±8%

Table 18. Summary of the results for samples.

	ASMOV	DSSim	Lily
P*	34% ± 10%	30% ± 8%	42% ± 10%
rrecall	18%	14%	17%

Table 19. Summary of the results for entire populations.

recall (*rrecall*) that is computed as ratio of the number of all correct correspondences (sum of all correct correspondences per one system) to the number of all correct correspondences found by any of systems (per all systems). This relative recall was computed over stratified random samples, so it is rather sample relative recall.

Discussion Although the ASMOV system achieves the highest result in two strata and the Lily system in the approximated precision P*, because of overlapping margins of errors we cannot say that a system outperforms another. In order to make approximated results more decisive we should take larger samples. Regarding relative recall, ASMOV achieves the highest value.

Evaluation based on reference alignments This is the classical evaluation method where the alignments from participants are compared against the *reference alignment*. So far we have built the reference alignment over five ontologies (cmt, confOf, ekaw, iasted, sigkdd, i.e. 10 alignments); we plan to cover the whole collection in the future. The decision about each correspondence was based on majority vote of three evaluators. In the case of disagreement among evaluators, the given correspondence was the subject of broader public discussion during the Consensus building workshop in order to find consensus and update the reference alignment, see the section (below) about the Evaluation based on the consensus of experts.

	t=0.2			t=0.5			t=0.7		
	P	R	F-meas	P	R	F-meas	P	R	F-meas
ASMOV	51.8%	38.6%	44.2%	72.2%	11.4%	19.7%	100.0%	6.1%	11.6%
DSSim	34.0%	57.9%	42.9%	34.0%	57.9%	42.9%	34.0%	57.9%	42.9%
Lily	43.2%	50.0%	46.3%	60.4%	28.1%	38.3%	66.7%	8.8%	15.5%

Table 20. Recall, precision and F-measure for three different thresholds

In Table 20, there are traditional *precision* (P), *recall* (R), and *F-measure* (F-meas) computed for three diverse thresholds (0.2, 0.5, and 0.7). As we have mentioned, these results are biased because the current reference alignment only covers a subset of all ontology pairs from the OntoFarm collection.

Discussion All systems achieve the highest F-measure for threshold 0.2, while the Lily system has the highest F-measure of 46.3%. The ASMOV system achieves the highest precision for each of three thresholds (51.8%, 72.2%, 100%) however it is at the expense of recall that is the lowest for each of three thresholds (38.6%, 11.4%, 6.1%). The highest recall (57.9%) was obtained by the DSSim system.

Evaluation based on data mining method This kind of evaluation is based on data mining, and the goal is to reveal non-trivial findings about the participating systems. These findings relate to the relationships between the particular system and features such as the confidence measure, validity, kinds of ontologies, particular ontologies, and *mapping patterns*. Mapping patterns have been introduced in [19]. For the purpose of our current experiment we extended detected mapping patterns with some patterns inspired by *correspondence patterns* [16] and with *error mapping patterns*.

Basically, mapping patterns are patterns dealing with (at least) two ontologies. These patterns reflect the *the structure of ontologies* on the one side, and on the other side they include correspondences between entities of ontologies. Initially, we discover some mapping patterns such as occurrences of some complex structures in the participants results. They are neither the result of a deliberate activity of humans, nor they are a priori 'desirable' or 'undesirable'. Here are three such mapping patterns between concepts:

- MP1 (Parent-child triangle): it consists of an equivalence correspondence between A and B and an equivalence correspondence between A and a child of B, where A and B are from different ontologies.
- MP2 (Mapping along taxonomy): it consists of simultaneous equivalence correspondences between parents and between children.
- MP3 (Sibling-sibling triangle): it consists of simultaneous correspondences between class A and two sibling classes C and D where A is from one ontology and C and D are from another ontology.

This year, we added three mapping patterns inspired by correspondence patterns [16]:

- MP4: it is inspired by the 'class by attribute' correspondence pattern, where the class in one ontology is restricted to only those instances having a particular value for a a given attribute/relation.
- MP5: it is inspired by the 'composite' correspondence pattern. It consists of a class-to-class equivalence correspondence and a property-to-property equivalence correspondence, where classes from the first correspondence are in the domain or in the range of properties from the second correspondence.
- MP6: it is inspired by the 'attribute to relation' correspondence pattern where a datatype and an object property are aligned as an equivalence correspondence.

Furthermore, there are error mapping patterns, which can disclose incorrect correspondences:

- MP7: it is the variant of MP5 ‘composite pattern’. It consists of an equivalence correspondence between two classes and an equivalence correspondence between two properties, where one class from the first correspondence is in the domain and another class from that correspondence is in the range of equivalent properties, except the case where domain and range is the same class.
- MP8: it consists of an equivalence correspondence between A and B and an equivalence correspondence between a child of A and a parent of B where A and B are from different ontologies. It is sometimes reffered to as criss-cross pattern.
- MP9: it is the variant of MP3, where the two sibling classes C and D are disjoint.

	MP1	MP2	MP3	MP4	MP5	MP6	MP7	MP8	MP9
ALL	0/543/0	255/146/115	0/527/0	261/828/354	467/115/585	132/115/151	0/6/13	0/7/4	0/165/0
REF	0/70/0	39/19/17	0/58/0	35/88/35	51/6/29	1/2/3	0/0/0	0/3/0	0/27/0

Table 21. Occurrences of mapping patterns in participants results.

In Table 21 there are numbers of correspondences found by each system (ASMOV/DSSim/Lily) that belong to a particular mapping pattern. The row ‘ALL’ relates to all equivalence correspondences delivered by participants with confidence measure higher than 0.0 (1540/1950/1744). The row ‘REF’ relates to all equivalence correspondences delivered by participants with confidence measure higher than 0.0 for pairs of ontologies for which there exists the reference alignment (182/194/132).

For the *data-mining analysis* we employed the *4ft-Miner* procedure of the *LISp-Miner* data mining system[21] for mining of *association rules*. For the sake of brevity we mention a few examples of interesting *association hypotheses* discovered[22]:

- In correspondences with low confidence measure [0,0.4) the ASMOV system comes 1.2 times more often with incorrect correspondences for *cmt* and *confOf* pair of ontologies than all systems with such (incorrect) correspondences for those two ontologies with all confidence measures (on average).
- The Lily system outputs almost three times more often correspondences that belong to the mapping pattern MP7 than do all systems (on average).
- In correspondences with low confidence measure [0,0.4) the Lily system comes 1.2 times more often with correct correspondences for pairs of ontologies with *iasted* ontology than all systems with such (correct) correspondences for those pairs of ontologies with all confidence measures (on average).

Discussion The abovementioned hypotheses disclose potentially interesting relationships for the developers of systems. By Table 21 (particularly numbers for MP7, MP8, and mainly for MP9) we could say that application of error mapping patterns

[21] `http://lispminer.vse.cz/`

[22] For association hypotheses with confidence measures we used REF correspondences, otherwise we used ALL correspondences.

would improve the systems' performance (for Lily to some degree and especially for DSSim) in terms of precision, while the results of the ASMOV system do not contain any instances of error mapping patterns due to its semantic verification phase.

Evaluation based on alignment incoherence Several ways to measure the incoherence of an alignment have been proposed in [13]. In the following we focus on the maximum cardinality measure m^t_{card} which has been introduced as revision based measure. The m^t_{card} measure compares the number of correspondences which have to be removed to arrive at a coherent subset with the number of all correspondences in the alignment. The conference ontologies are well suited for an analysis of alignment incoherence since most of them contain negation as well as different kinds of restrictions exploiting the range of OWL-DL expressivity.

Due to practical considerations we decided to modify the approach with respect to two aspects. First, we observed that many logical problems induced by an alignment are related to properties. Therefore, we applied a different definition of incoherence taking property unsatisfiability into account. We defined an ontology to be incoherent whenever there exists an unsatisfiable concept or property. This extends the classical approach in which ontology incoherence depends only on the unsatisfiability of concepts (see for example [14]). Second, we observed that matching object properties on datatype properties might be an appropriate way to cope with semantic heterogeneity. Nevertheless, such a correspondence would directly result in an incoherent alignment based on the direct natural translation of a correspondence as axiom. Therefore, we used a slightly modified variant of the natural translation and translated each correspondence between properties R_1 and R_2 into an axiom $\exists R_1.\top \equiv \exists R_2.\top$ (we only considered equivalence correspondences).

System	Alignments	Coherent	Mean	Median
ASMOV	44 (1010)	8	0.135	0.14
Lily	45 (851)	9	0.138	0.145
DSSim	45 (769)	3	0.206	0.166

Table 22. Number of evaluated alignments (and total of correspondences), number of coherent alignments, mean and median for the maximum cardinality measure..

In our experimental evaluation we considered only a subset of 10 ontologies and evaluated the alignments between all possible pairs. We excluded five ontologies (Cocus, Confious, Iasted, Paperdyne and OpenConf) because we only focused on alignments submitted by each participant and encountered reasoning problems for some of these ontologies. Table 22 summarizes the main results. First of all we notice that only a small fraction of submitted alignments is coherent. For ASMOV and Lily 18% resp. 20% of the evaluated alignments were coherent, while DSSim generated only 7% coherent alignments. We also computed the mean of the m^t_{card} measure over all analyzed alignments. We observe that ASMOV and Lily generate alignments with a lower degree of incoherence (0.135 and 0.138) compared to DSSim (0.206).

The distribution of measured values additionally supports our first impression. Figure13 shows the second and third quartile as well as the median of the values measured via m^t_{card}. While Lily and especially ASMOV found a way to prevent highly incoherent alignments, 25% of the alignments generated by DSSim have a degree of incoherence greater or equal than 0.288. For each of these alignments there are logical reasons to remove at least one-fourth of its correspondences. The differences between ASMOV, Lily and DSSim revealed by our incoherence analysis fits with the differences we reported on the occurence of the error mapping patterns MP7 to MP9.

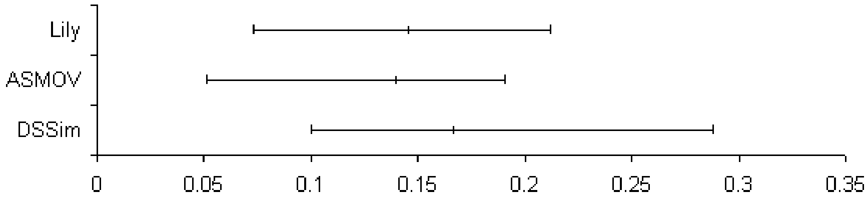


Fig. 13. Distribution of m^t_{card} values, depicting second quartile, median, and third quartile.

Discussion Some of the participants implemented a component to debug or validate generated alignments, namely ASMOV and Lily. To our knowledge these debugging techniques are based on detecting certain structural patterns in correspondence pairs (MP7 to MP9 can be seen as examples of such patterns). Although these strategies cannot ensure the coherence of an alignment, such an approach is nevertheless an efficient way to avoid full-fledged reasoning while increasing the degree of coherence. Taking alignment coherence into account can be a useful guide for improving the results of a matching system and our results suggest that there is still room for improvement.

Evaluation based on consensus of experts During so-called Consensus building workshop we discussed 5 controversial correspondences. The main goal of this discussion among experts was to find consensus about those correspondences and track arguments against and favour. This session ratified insights from previous years and disclosed that finding consensus is time-consuming and not an easy activity however doable. Some other relevant topics were raised. For instance, open-world assumption vs. closed-world assumption was considered as an important factor for understanding the description of entities in ontologies. The need for expressive alignments also arouse for expressing complex correspondences combining several elements (classes or properties). The reached consensus is captured in the reference alignment and discussion can be further proceed in the blog[23].

[23] `http://keg.vse.cz/oaei/`

10.3 Conclusion

In conclusion, we evaluated participant results from diverse perspectives via five distinct evaluation methods. For next year of this track, we also plan to evaluate subsumption correspondences and further extend the reference alignment. Based on the participants' feedback we changed ontologies from the OntoFarm collection in order to be OWL DL compliant for the next year of the conference track.

11 Lesson learned and suggestions

The lessons learned for this year are relatively similar to those of previous years. But there remain lessons not really taken into account that we identify with an asterisk (*). We reiterate those lessons that still apply with new ones:

A) Unfortunately, we have not been able to maintain the better schedule of last year. With the schedule reduced by one month (thus in overall having about 3 months), it is very difficult to run OAEI.

B) Some of the best systems of last year did not enter. The invoked reasons were: not enough time and/or no improvement in the systems. This pleads for continous instead of yearly evaluation.

C) The trend that there are more matching systems able to enter such an evaluation seems to slow down. However, the number of tracks the existing systems are able to consider still very encouraging for the progress of the field.

D) We can confirm that systems that enter the campaign for several times tend to improve over years.

E*) The benchmark test case is not discriminant enough between systems. It is still useful for evaluating the strengths and weaknesses of algorithms but does not seem to be sufficient anymore for comparing algorithms. We have improved tests this year, while preserving comparability with previous years, but more is required, in particular in automatic test generation.

F) We have had more proposals for test cases this year. 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 becomes easier to perform the evaluation.

G) There are now test cases where non equivalence-only alignments matter and there are systems, e.g., ASMOV, Aroma, TaxoMap, which are able to deliver such alignments. We thus intent to have such a test case next year. The discussion about instance matching tests has also aroused.

H) The robustness of evaluation tools make that, like last year, we had very few syntactic problems this year. However, it seems that many matchers are too dependent on particular operating systems and still many ones do not deal correctly with ontology URIs (see the Error cells in Table 3).

I) The partition between systems able to deal with large ontologies and systems unable to do it seems to be transforming gradually: systems seem to be able to perform more tasks. However, this requires an important amount of manpower.

12 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, especially with expressive ontologies;
- 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);
- Developing a definition of test hardness.

Of course, these are only suggestions that will be refined during the coming year, see [17] for a detailed discussion on the ontology matching challenges.

13 Conclusions

This year we had less systems overall entering the evaluation campaign with still a significant number of systems. It seems however that they entered more tests individually (50 last year overall against 48 this year), so systems seem to be more up to the challenge.

As noticed the previous years, systems which do not enter for the first time are those which perform 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).

All participants have provided description of their systems and their experience in the evaluation. These OAEI papers, like the present one, have not been peer reviewed. However, they are full contributions to this evaluation exercise and reflect the hard work and clever insight people put in the development of participating systems. 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. Sometimes participants offer alternate evaluation results.

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 campaign. 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 the following papers.

We are grateful to Martin Ringwald and Terry Hayamizu for providing the reference alignment for the anatomy ontologies.

Thanks to Andrew Bagdanov, Aureliano Gentile, Gudrun Johannsen (Food and Agriculture Organization of the United Nations) for evaluating the FAO task. We also thank the teams of Agricultural Organization of the United Nations (FAO) for allowing us to use their ontologies. Caterina Caraciolo and Jérôme Euzenat have been partially supported by the European integrated project NeOn (IST-2005-027595).

We are grateful to Henk Matthezing, Lourens van der Meij and Shenghui Wang who have made crucial contributions to implementation and reporting for the Library track. The evaluation at KB could not have been possible without the commitment of Yvonne van der Steen, Irene Wolters, Maarten van Schie, and Erik Oltmans.

We thank Chris Bizer, Fabian Suchanec and Jens Lehman for their help with the DBPedia dataset. We also thank Willem van Hage for his advices. We gratefully acknowledge the Dutch Institute for Sound and Vision for allowing us to use the GTAA.

We are grateful to Peter Bartoš (Brno University of Technology, CZ) for participating in creation of partial reference alignment for the conference track. In addition, Ondřej Šváb-Zamazal and Vojtěch Svátek were supported by the IGA VSE grant no.20/08 "Evaluation and matching ontologies via patterns".

We also thank the other members of the Ontology Alignment Evaluation Initiative Steering committee: Wayne Bethea (John Hopkins University, USA), Alfio Ferrara (Università degli Studi di Milano, Italy), Lewis Hart (AT&T, USA), Tadashi Hoshiai (Fujitsu, Japan), Todd Hughes (DARPA, USA), Yannis Kalfoglou (University of Southampton, UK), John Li (Teknowledge, USA), Miklos Nagy (The Open University (UK), Natasha Noy (Stanford University, USA), Yuzhong Qu (Southeast University (China), York Sure (University of Karlsruhe, Germany), Jie Tang (Tsinghua University (China), Raphaël Troncy (CWI, Amsterdam, The Netherlands), Petko Valtchev (Université du Québec à Montréal, Canada), and George Vouros (University of the Aegean, Greece).

References

1. Zharko Aleksovski, Warner ten Kate, and Frank van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In *Proceedings of the ISWC international workshop on Ontology Matching*, pages 13–24, Athens (GA US), 2006.
2. Ben Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proceedings of the K-Cap workshop on Integrating Ontologies*, Banff (CA), 2005.
3. Oliver Bodenreider, Terry F. Hayamizu, Martin Ringwald, Sherri De Coronado, and Songmao Zhang. Of mice and men: Aligning mouse and human anatomies. In *Proceedings of the American Medical Informatics Association (AIMA) Annual Symposium*, pages 61–65, 2005.
4. Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In *Proceedings of the K-Cap workshop on Integrating Ontologies*, pages 25–32, Banff (CA), 2005.
5. Jérôme Euzenat. An API for ontology alignment. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pages 698–712, Hiroshima (JP), 2004.
6. Jérôme Euzenat, Malgorzata Mochol, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2006. In Pavel Shvaiko, Jérôme Euzenat, Natalya Noy, Heiner Stuckenschmidt, Richard Benjamins, and Michael Uschold, editors, *Proceedings of the ISWC international workshop on Ontology Matching, Athens (GA US)*, pages 73–95, 2006.
7. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, Heidelberg (DE), 2007.

8. Jérôme Euzenat, Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2007. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Bin He, editors, *Proceedings of the 2nd ISWC international workshop on Ontology Matching, Busan (KR)*, pages 96–132, 2007.
9. Fausto Giunchiglia, Mikalai Yatskevich, Paolo Avesani, and Pavel Shvaiko. A large scale dataset for the evaluation of ontology matching systems. *The Knowledge Engineering Review Journal*, (24(2)), 2009, to appear.
10. Ryutaro Ichise, Masahiro Hamasaki, and Hideaki Takeda. Discovering relationships among catalogs. In *Proceedings of the 7th International Conference on Discovery Science*, pages 371–379, Padova (IT), 2004.
11. Ryutaro Ichise, Hideaki Takeda, and Shinichi Honiden. Integrating multiple internet directories by instance-based learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 22–28, Acapulco (MX), 2003.
12. Antoine Isaac, Henk Matthezing, Lourens van der Meij, Stefan Schlobach, Shenghui Wang, and Claus Zinn. Putting ontology alignment in context: Usage scenarios, deployment and evaluation in a library case. In *Proceedings of the 5th European Semantic Web Conference (ESWC)*, pages 402–417, Tenerife (ES), 2008.
13. Christian Meilicke and Heiner Stuckenschmidt. Incoherence as a basis for measuring the quality of ontology mappings. In *Proceedings of the 3rd ISWC international workshop on Ontology Matching*, pages 1–12, Karlsruhe (DE), 2008.
14. Guilin Qi and Anthony Hunter. Measuring incoherence in description logic-based ontologies. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, pages 381–394, Busan (KR), 2007.
15. Marta Sabou, Mathieu d'Aquin, and Enrico Motta. Using the semantic web as background knowledge for ontology mapping. In *Proceedings of the ISWC international workshop on Ontology Matching*, pages 1–12, Athens (GA US), 2006.
16. Francois Scharffe and Dieter Fensel. Correspondence patterns for ontology alignment. In *Proceedings of the 16th International Conference on Knowledge Acquisition, Modeling and Management (EKAW)*, pages 83–92, Acitrezza (IT), 2008.
17. Pavel Shvaiko and Jérôme Euzenat. Ten challenges for ontology matching. In *Proceedings of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, pages 1164–1182, Monterrey (MX), 2008.
18. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the ISWC workshop on Evaluation of Ontology-based tools (EON)*, Hiroshima (JP), 2004.
19. Ondrej Svab, Vojtech Svatek, and Heiner Stuckenschmidt. A study in empirical and 'casuistic' analysis of ontology mapping results. In *Proceedings of the 4th European Semantic Web Conference (ESWC)*, pages 655–669, Innsbruck (AU), 2007.
20. Willem Robert van Hage, Antoine Isaac, and Aleksovski, Zharko. Sample evaluation of ontology matching systems. In *Proceedings of the ISWC workshop on Evaluation of Ontologies and Ontology-based tools*, pages 41–50, Busan (KR), 2007.

Roma, Grenoble, Tokyo, Amsterdam, Trento, Mannheim, and Prague, December 2008

Alignment Results of Anchor-Flood Algorithm for OAEI-2008

Md. Hanif Seddiqui and Masaki Aono

Toyohashi University of Technology, Aichi, Japan
hanif@kde.ics.tut.ac.jp, aono@ics.tut.ac.jp

Abstract. Our proposed algorithm called *Anchor-Flood algorithm*, starts off with *anchors*. It gradually explores concepts by collecting neighbors in concept taxonomy, thereby taking advantage of *locality of reference* in the graph data structure. Then local alignment process runs over the collected small blocks of concepts. The process is repeated for the newly found aligned pairs. In this way, we can significantly reduce the computational time for the alignment as our algorithm concentrates on the aligned pairs and it resolves the scalability problem in ontology alignment over large ontologies. Through several experiments against OAEI-2008 datasets, we will demonstrate the results and the features of our Anchor-Food algorithm.

1 Presentation of the system

The Anchor-Flood algorithm is mainly designed targeting to align two large scale ontologies or one large scale and another small scale ontologies effectively. It does not compare an entity against all the entities in other ontology. The way of selecting the group of entities to be compared is the novelty of our algorithm. Our algorithm operates quite faster over large ontologies as observed in aligning anatomy ontologies and it is depicted in Table 2.

1.1 State, purpose, general statement

The purpose of our Anchor-Flood algorithm is basically ontology matching. However, we used our algorithm in patent mining system to classify a research abstract in terms of International Patent Classification (IPC). Containing mostly general terminologies leads classifying an abstract a formidable task. Automatic extracted taxonomy of related terms available in an abstract is aligned with the taxonomy of IPC ontology with our algorithm succesfully. We also start using the Anchor-Flood in the focus-oriented biomedical applications which generally contain very large ontologies.

To be specific, we only describe our Anchor-Flood algorithm and the results against OAEI 2008 datasets here. For more details, we refer the reader to our semantic website : http://www.kde.ics.tut.ac.jp/h̃anif. More elaborate information will be come out soon in our semantic technology geared website.

1.2 Specific techniques used

We implemented Anchor -Flood algorithm in java. Our algorithm contains preprocessing, adaptation module for OAEI 2008, the basic block of algorihtm and the local alignment process.

We created our own persistent model of ontology, as our algorithm requires optimal graph structure of concept taxonomy along with other non-trivial structural and simple lexical information. To collect the necessary information in repository, we use the ARP triple parser of jena module. Fig 1 shows the basic block of Anchor-Flood algorithm to comprehend easily. However, it has complex process of collecting small blocks of concepts and related properties dynamically.

As a part of preprocessing, we also normalize the lexical information and extract the derivative relations, like inherited restrictions etc.

The basic part of Anchor-Flood algorithm is depicted in Fig. 1. Starting off an anchor, Anchor-Flood algorithm collects neighboring concepts which includes super concepts, siblings and subconcepts of certain depth to form a pair of blocks across ontologies, as the neighbors of similar concepts might also be similar [5]. Local alignment process aligns concepts and their related properties based on lexical information [2, 7, 8], semantic information [4] and structural relations [1, 3, 4]. Found aligned pairs are considered for further processing. Hence, it burst out with a pair of aligned block in a compacked part of the ontologies, giving the taste of segmentation [6].

Multiple anchors from different part of ontologies confirm a fair collection of aligned pairs as a whole.

1.3 Adaptations made for the evaluation

The Anchor-Flood algorithm needs an anchor to start off. Therefore, we used another tiny program module, which is capable of extarcting some probable aligned pairs as anchors. The tiny program is attached inside along with our basic algorithm to produce a system. It uses lexical information and some statistical relational information to extract a small number of aligned pairs from different part of ontologies. The program is essentially small, simple and faster. We also removed the subsumption module of our algorithm to make it more faster.

1.4 Link to the system and parameters file

The version of Anchor-Flood for OAEI-2008 can be downloaded from our website: http://www.kde.ics.tut.ac.jp/h̃anif/res/anchor_flood.zip

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

The results for OAEI-2008 are available at our website: http://www.kde.ics.tut.ac.jp/ h̃anif/res/aflood.zip

2 Results

In this section, we describe the results of Anchor-Flood algorithm against the benchmark and anatomy ontologies provided by the OAEI 2008 campaign.

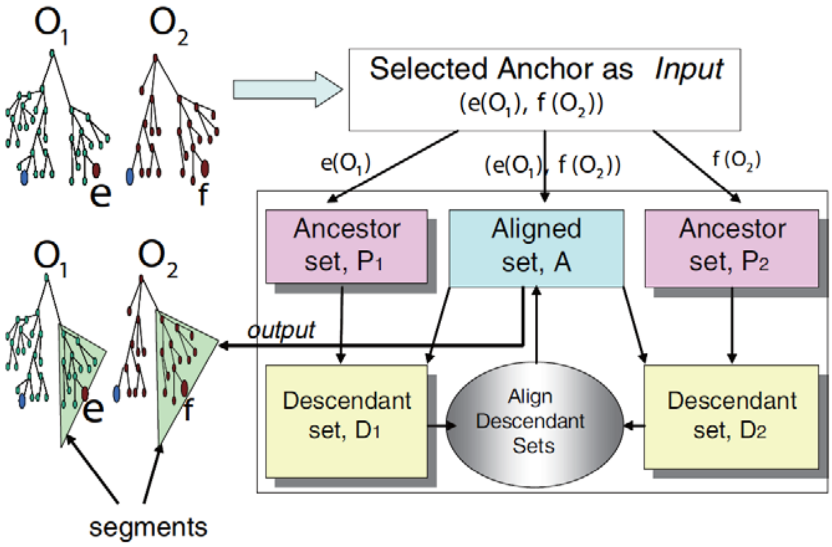


Fig. 1. The figure shows the process of Anchor-Flood algorithm where anchor is taken as an input to produce a segmented alignment. Multiple anchors produce a fair collection of aligned pairs as a whole.

2.1 Benchmark

On the basis of the nature, we can divide the benchmark dataset into five groups: #101-104, #201-210, #221-247, #248-266 and #301-304. We described the performance of our Anchor-Flood algorithm over each of the groups and depicted in . The overall summary over 1xx, 2xx and 3xx are also figured in .

#101-104 Table 1 shows that Anchor-Flood algorithm produces perfect precision and recall in this group.

#201-210 Although the lexical information of the ontologies are suppressed or modified, their structures remain quite similar. Therefore traversing the structure with taxonomy and relation works better for this group.

#221-247 The structures of the candidate ontologies are altered. However, the dynamic block collector of our Anchor-Flood algorithm can collect concepts and properties as the ontologies are small in size. Therefore, it can still produce good precision and recall.

#248-266 This is the most difficult group for our Anchor-Flood algorithm, as the structure and the lexical information altered significantly. However, the subgroups with xxx-2 through xxx-8 are seemingly easier to align.

Table 1. Summary of the average precision, recall and total elapsed time.

#	Prec.	Rec.	F-Measure	Total Time (sec)
1xx	1.00	1.00	1.00	2.67
2xx	0.93	0.69	0.79	87.08
3xx	0.88	0.79	0.83	3.43
Average	0.93	0.70	0.77	

#301-304 Anchor-Flood algorithm in this group works well even after removing the subsumption module from our main algorithm. Both structural and lexical analysis works well in this group.

2.2 Anatomy

In this test, the real world cases of anatomy for Adult Mouse Anatomy (2744 classes) and NCI Thesaurus (3304 classes) for human anatomy are included. These are relatively large compared to benchmark ontologies. The actual effectivity of our Anchor-Flood algorithm shows with faster operational time. It collects 1187 aligned pairs within only 1.09 minutes in our Core2 Duo 2.4MHz processor with 2GB of memory. Table 2 shows the summary of the performance on the anatomy task.

Table 2. The Anchor-Flood algorithm collects aligned pairs from anatomy ontologies quickly. The Table shows the brief summary of the output.

Total Aligned Pair	Required Time (m)
1187	1.09

3 General comments

In this section, we want to introduce comments on the results of Anchor-Flood algorithm and the way to improve the proposed system

3.1 Comments on the results

The main strength of our Anchor-Flood algorithm is the way of minimizing the comparisons between entities, which leads enhancement in performance. It has some better scope in the field of ontology versioning of small specific domain ontologies comparing with other large ontologies.

The weak points are: it has still rooms of improving alignments based on axioms, semantic similarity, and structures of ontologies.

3.2 Discussions on the way to improve the proposed system

The subsumption module of our algorithm takes much time. Our next plan is to improve the alignments on the basis of axioms and structures and improving the subsumption module as well.

4 Conclusion

Ontology matching is very important part of establishing interoperability among semantic application as the core of every semantic application is ontology. We implemented faster algorithm to align specific interrelated parts across ontologies, which gives the flavor of segmentation. Pair of segmented aligned part across ontology can be used versioning ontologies, e.g. cancer ontology versioning with the general diseases ontology. The anatomical ontology matching shows the effectiveness of our Anchor-Flood algorithm. Moreover, the experimental experience in the OAEI 2008 campaign will influence us building comprehensive ontology matching system removing the limitations of our algorithm in the future.

Acknowledgements

This study was partially supported by Global COE Program gFrontiers of Intelligent Sensingh from Japan's Ministry of Education, Culture, Sports, Science and Technology.

References

1. P. Bouquet, L. Serafini, S. Zanobini, Semantic Coordination: A New Approach and an Application, Proceedings of the 2nd International Semantic Web Conference (ISWC2003), Sanibel Island, Florida, USA (2003) 130–145.
2. J. Euzenat, P. Valtchev, Similarity-based Ontology Alignment in OWL-Lite, Proceedings of the 16th European Conference on Artificial Intelligence (ECAI2004), Valencia, Spain (2004) 333–337.
3. F. Giunchiglia, P. Shaiko, Semantic Matching, The Knowledge Engineering Review 18 (03) (2004) 265–280.
4. F. Giunchiglia, P. Shvaiko, M. Yatskevich, S-Match: an Algorithm and an Implementation of Semantic Matching, Proceedings of the 1st European Semantic Web Symposium (ESWS2004), Heraklion, Greece (2004) 61–75.
5. S. Melnik, H. Garcia-Molina, E. Rahm, Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching, Proceedings of the 18th International Conference on Data Engineering (ICDE 2002), San Jose, CA (2002) 117–128.
6. J. Seidenberg, A. Rector, Web Ontology Segmentation: Analysis, Classification and Use, Proceedings of the 15th International Conference on World Wide Web (WWW2006), Edinburgh, Scotland (2006) 13–22.
7. G. Stoilos, G. Stamou, S. Kollias, A String Metric for Ontology Alignment, Proceedings of the 4th International Semantic Web Conference (ISWC2005), Galway, Ireland (2005) 623–637.
8. W. E. Winkler, The State of Record Linkage and Current Research Problems, Technical report, Statistical Research Division, U.S. Census Bureau, Washington 1999.

Appendix: Raw Results

The tests are carried out on an Intel Core 2 Duo 2.4MHz desktop machine with 2GB DDR2 memory under Windows XP Professional operating system and Java 1.6.0_02 compiler

Matrix of Results

The following table contains the results of Anchor-Flood algorithm in the benchmark test. The table includes precision (Prec.), recall (Rec.) and processing time. The processing time includes construction of model, execution of algorithm to produce aligned pairs and writing the results into a .rdf file.

#	Prec.	Rec.	F-Measure	Time (sec)
101	1.00	1.00	1.00	0.94
101	1.00	1.00	1.00	0.94
103	1.00	1.00	1.00	0.89
104	1.00	1.00	1.00	0.84
201	0.96	0.84	0.90	0.92
201-2	1.00	0.86	0.92	0.91
201-4	1.00	0.79	0.88	0.89
201-6	0.98	0.91	0.94	0.89
201-8	0.95	0.58	0.72	0.91
202	1.00	0.78	0.88	0.86
202-2	1.00	0.86	0.92	0.86
202-4	1.00	0.92	0.96	0.84
202-6	1.00	0.88	0.94	0.86
202-8	0.91	0.42	0.57	0.84
203	1.00	1.00	1.00	0.86
204	1.00	0.97	0.98	0.91
205	0.89	0.57	0.69	0.94
206	0.96	0.84	0.90	0.92
207	0.96	0.84	0.90	0.95
208	1.00	0.96	0.98	0.89
209	0.91	0.49	0.64	0.92
210	0.96	0.80	0.87	0.86
221	1.00	1.00	1.00	0.91
222	1.00	1.00	1.00	0.88
223	1.00	1.00	1.00	0.95
224	1.00	1.00	1.00	0.81
225	1.00	1.00	1.00	0.88
228	1.00	1.00	1.00	0.72
230	0.94	1.00	0.97	0.88
231	1.00	1.00	1.00	0.88

232	1.00	1.00	1.00	0.83
233	1.00	1.00	1.00	0.74
236	1.00	1.00	1.00	0.74
237	1.00	1.00	1.00	0.81
238	1.00	1.00	1.00	0.91
239	0.97	1.00	0.98	0.70
240	0.97	1.00	0.98	0.8 0
241	1.00	1.00	1.00	0.69
246	0.97	1.00	0.98	0.74
247	0.97	1.00	0.98	0.84
248	0.50	0.12	0.19	0.84
248-2	0.99	0.78	0.87	0.81
248-4	0.97	0.64	0.77	0.84
248-6	0.94	0.51	0.66	0.86
248-8	0.74	0.32	0.45	0.92
249	0.92	0.24	0.38	0.86
249-2	1.00	0.86	0.92	0.88
249-4	1.00	0.90	0.95	0.84
249-6	0.93	0.58	0.71	0.88
249-8	0.95	0.77	0.85	0.84
250	1.00	0.33	0.50	0.91
250-2	1.00	0.85	0.92	0.91
250-4	1.00	0.73	0.84	0.84
250-6	1.00	0.64	0.78	0.88
250-8	1.00	0.48	0.65	0.84
251	1.00	0.32	0.48	0.91
251-2	1.00	0.85	0.92	0.91
251-4	1.00	0.76	0.86	0.92
251-6	0.97	0.62	0.76	0.91
251-8	0.92	0.47	0.62	0.92
252	0.80	0.08	0.15	0.92
252-2	0.97	0.79	0.87	0.97
252-4	0.97	0.79	0.87	0.92
252-6	0.98	0.80	0.88	0.95
252-8	0.98	0.80	0.88	0.92
253	0.46	0.11	0.18	0.86
253-2	0.99	0.78	0.87	0.86
253-4	0.99	0.68	0.81	0.88
253-6	0.93	0.52	0.67	0.84
253-8	0.79	0.34	0.48	0.86
254	1.00	0.27	0.43	0.75
254-2	1.00	0.82	0.90	0.75
254-4	1.00	0.70	0.82	0.73

254-6	1.00	0.61	0.76	0.77
254-8	1.00	0.42	0.59	0.73
257	0.50	0.06	0.11	0.73
257-2	1.00	0.82	0.90	0.70
257-4	0.95	0.64	0.76	0.72
257-6	0.88	0.45	0.60	0.72
257-8	0.82	0.27	0.41	0.72
258	1.00	0.31	0.47	0.84
258-2	1.00	0.86	0.92	0.89
258-4	1.00	0.76	0.86	0.88
258-6	0.97	0.62	0.76	0.86
258-8	0.91	0.46	0.61	0.84
259	0.11	0.01	0.02	0.84
259-2	0.98	0.80	0.88	0.91
259-4	0.98	0.80	0.88	0.92
259-6	0.98	0.80	0.88	0.89
259-8	0.97	0.79	0.87	0.91
260	0.92	0.38	0.54	0.72
260-2	0.96	0.86	0.91	0.75
260-4	0.96	0.76	0.85	0.74
260-6	0.95	0.66	0.78	0.75
260-8	0.94	0.55	0.69	0.77
261	0.82	0.27	0.41	0.77
261-2	0.96	0.82	0.88	0.77
261-4	0.97	0.85	0.91	0.75
261-6	0.97	0.85	0.91	0.77
261-8	0.96	0.82	0.88	0.83
262-2	1.00	0.79	0.88	0.74
262-4	1.00	0.61	0.76	0.72
262-6	1.00	0.42	0.59	0.73
262-8	1.00	0.21	0.35	0.78
265	0.50	0.07	0.12	0.70
266	0.25	0.03	0.05	0.72
301	0.98	0.82	0.89	0.81
302	0.83	0.60	0.70	0.80
303	0.75	0.79	0.77	0.94
304	0.95	0.95	0.95	0.88
Total	0.93	0.70	0.80	91.45

AROMA results for OAEI 2008

Jérôme David[1]

INRIA Rhône-Alpes, Montbonnot Saint-Martin, France
Jerome.David-at-inrialpes.fr

Abstract. This paper presents the results obtained by AROMA for its first participation to OAEI. AROMA is an hybrid, extensional and asymmetric ontology alignment method which makes use of the association paradigm and a statistical interstingness measure, the implication intensity.

1 Presentation of AROMA

1.1 State, purpose, general statement

AROMA is an hybrid, extensional and asymmetric matching approach designed to find out relations (equivalence and subsumption) between entities issued from two textual taxonomies (web directories or OWL ontologies). Our approach makes use of the association rule paradigm [Agrawal *et al.*, 1993], and a statistical interestingness measure used in this context. AROMA relies on the following assumption: *An entity A will be more specific than or equivalent to an entity B if the vocabulary (i.e. terms and also data) used to describe A, its descendants, and its instances tends to be included in that of B.*

1.2 Specific techniques used

AROMA is divided into three successive main stages: (1) The pre processing stage allows to represent each entity (classes and properties) by a set of terms, (2) the second stage consists of the discovery of association rules between entities, and finally (3) the post processing stage aims to clean and enhance the alignment.

The first stage constructs a set of relevant terms and/or datavalues for each entity. To do this, we extract the vocabulary of entities from their annotations and individual values with the help of single and binary term extractor applied to stemmed text.

The second stage of AROMA discovers the subsumption relations by using the association rule model and the implication intensity measure [Gras *et al.*, 2008]. In the context of AROMA, an association rule $a \rightarrow b$ represents a quasi-implication (i.e. an implication allowing some counter-examples) from the vocabulary of entity a into the vocabulary of the entity b. Such a rule could be interpreted as a subsumption relation from the antecedent entity toward the consequent one. For example, the binary rule $car \rightarrow vehicle$ could be interpret: "The concept car is more specific than the concept $vehicle$". The rule extraction algorithm takes advantage of the partial order structure provided by the subsumption relation, and a property of the implication intensity for pruning the search space.

The last stage concerns the post processing of the association rule set. It performs the following tasks:

- the deduction of equivalence relations,
- the suppression of cycles in the alignment graph,
- the suppression of the redundant correspondences,
- the selection of the best correspondence for each entity (the alignment is an injective function),
- the enhancement of the alignment by using a string similarity -based matcher (Jaro-Winkler similarity) and previously discovered correspondences.

For more details, the reader should refer to [David *et al.*, 2007; David, 2007].

1.3 Link to the system and parameters file

The version of AROMA used for OAEI 2008 is available at:
`http://www.inrialpes.fr/exmo/people/jdavid/oaei2008/AROMA_oaei2008.jar`.

The source code is available at :
`http://www.inrialpes.fr/exmo/people/jdavid/oaei2008/AROMAsrc_oaei2008.jar`

For align two ontologies use the following command line:

```
java -jar AROMA_oaei2008.jar onto1.owl onto2.owl
```

The resulting alignment is provided on the standard output in the alignment format.

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

`http://www.inrialpes.fr/exmo/people/jdavid/oaei2008/results_AROMA_oaei2008.zip`

2 Results

We participated to benchmark, anatomy and fao tests. We used the same configuration of AROMA for all tests. We did not have scaling problems. We only comment benchmark results because we do not have the results on the other tests. We also discuss why we did not participate to directory and mldirectory tests.

2.1 benchmark

Since AROMA only relies on textual information, it obtains bad recall values when the alterations affect all text annotations both in the class/property descriptions and in their individual/property values. AROMA seems do be not influenced by structural alterations (222-247). On these tests, AROMA favors high precision values in comparison to recall values.

2.2 anatomy

On anatomy test, we do not use any particular knowledge about biomedical domain. Anatomy ontologies use their own annotation properties. We have made some adaptations in order to deal with these annotations.

In terms of precision and recall, AROMA performs worse than the label equality matcher. In particular, it obtains quite low recall value. Nevertheless, it discovers 30% of non-trivial correspondences that are not found by the term equality matcher.

Since AROMA takes benefits of the subsumption relation for pruning the search space, it runs quite fast. This pruning feature partially explains the low recall value obtained by AROMA on the anatomy test.

2.3 fao

We do not make any adaptation for this test. Fao ontologies (as anatomy) use their own, as a consequence, some textual data were not taken into account by AROMA.

On this test, AROMA also obtains low recall value and some results have not been evaluated due to the lack of returned correspondences.

2.4 directory

The two large directories (given previous years but not this year) are divided into very small sub directories. AROMA cannot align such very small directories because our method is based on a statistical measure and then it needs some large amount of textual data. However, AROMA discovers correspondences when it is applied to the complete directories. It would be interesting to reintroduce these large taxonomies for the next editions.

2.5 mldirectory

AROMA only relies on common textual data shared by ontologies to be align and it does not use multi-lingual resources. As a consequence, it does not work with this kind of tests.

3 General comments on AROMA

Even if results are quite good on benchmark track, alignments provided by AROMA have quite low recall values. This is partially due to the pruning strategy used. To overcome this drawback, we used a syntactical matcher in order to augment alignments. Even if it performs well on benchmarks ontologies, this matcher seems not to be very efficient on real cases (fao and anatomy).

4 Conclusion

For its first participation to OAEI, AROMA passed the benchmark, FAO and Anatomy tests. We do not have any scaling problem with these tests.

The results on benchmarks shows that AROMA is dependent on the amount of textual information available and it has bad results when both labels and comments are suppressed. However, AROMA is not very influenced by structural alterations. On anatomy track, AROMA has good runtimes but lacks in terms of recall. FAO track corroborates this drawback.

References

[Agrawal *et al.*, 1993] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM Press, 1993.

[David *et al.*, 2007] Jérôme David, Fabrice Guillet, and Henri Briand. Association rule ontology matching approach. *International Journal on Semantic Web and Information Systems*, 3(2):27–49, 2007.

[David, 2007] Jérôme David. *AROMA : une méthode pour la découverte d'alignements orientés entre ontologies à partir de règles d'association*. PhD thesis, Université de Nantes, 2007.

[Gras *et al.*, 2008] Régis Gras, Einoshin Suzuki, Fabrice Guillet, and Filippo Spagnolo, editors. *Statistical Implicative Analysis, Theory and Applications*, volume 127 of *Studies in Computational Intelligence*. Springer, 2008.

ASMOV: Results for OAEI 2008

Yves R. Jean-Mary[1], Mansur R. Kabuka [1,2]

[1] INFOTECH Soft, 9200 South Dadeland Blvd, Suite 620, Miami, Florida, USA 33156
[2] University of Miami, Coral Gables, Florida, USA 33124
{reggie, kabuka}@infotechsoft.com

Abstract. The Automated Semantic Mapping of Ontologies with Validation (ASMOV) algorithm for ontology alignment was one of the top performing algorithms in the 2007 Ontology Alignment Evaluation Initiative (OAEI). In this paper, we present a brief overview of the algorithm and its improvements, followed by an analysis of its results on the 2008 OAEI tests.

1 Presentation of the System

In recent years, ontology alignment has become popular in solving interoperability issues across heterogonous systems in the semantic web. Though many techniques have emerged from the literature [1], the distinction between them is accentuated by the manner in which they exploit the ontology features ASMOV, an algorithm that automates the ontology alignment process while optionally accepting feedback from a user, uses a weighted average of measurements of similarity along four different features of ontologies, and performs semantic validation of resulting alignments. A more complete description of ASMOV is presented in [3].

1.1 State, Purpose, General Statement

ASMOV is an automatic ontology matching tool which has been designed in order to facilitate the integration of heterogeneous systems, using their data source ontologies. The current ASMOV implementation produces mappings between concepts, properties, and individuals, including mappings from object properties to datatype properties and vice versa.

1.2 Specific Techniques Used

The ASMOV algorithm iteratively calculates the similarity between entities for a pair of ontologies by analyzing four features: lexical description (id, label, and comment), external structure (parents and children), internal structure (property restrictions for concepts; types, domains, and ranges for properties; data values for individuals), and individual similarity. The measures obtained by comparing these four features are combined into a single value using a weighted sum in a similar manner to [2]. These weights have been optimized based on the OAEI 2008 benchmark test results.

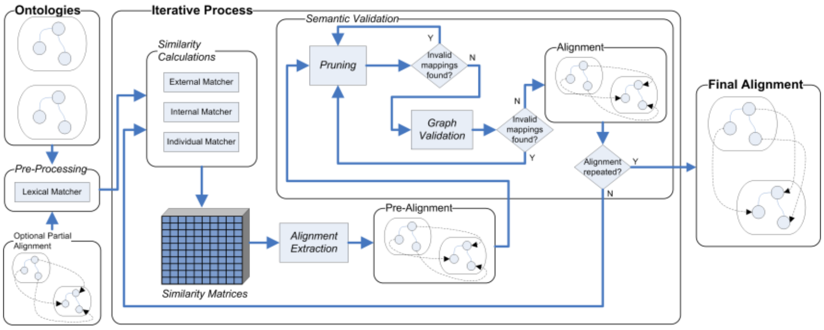


Fig. 1. The ASMOV Mapping Process

Fig. 1 illustrates the fully automated ASMOV mapping process, which has been implemented in Java. In the pre-processing phase, the ontologies are loaded into memory using the Jena ARP parser [4] and ASMOV's ontology modeling component. A thesaurus is then used in order to calculate the lexical similarities between each pair of concepts, properties and individuals. ASMOV can be configured to use either the UMLS Metathesaurus [5] or WordNet [6] in order to derive the similarity measures. A user can also opt to not use a thesaurus; in that case, a text matching algorithm is used to compute the lexical distance.

Following this, the similarities between pairs of entities along the external structure, internal structure, and individual dimensions are calculated, and an overall similarity measure (or confidence value) is stored in three two-dimensional matrices, one each for concepts, properties, and individuals. From these similarity matrices, a pre-alignment is obtained by selecting the entity from one ontology with the highest confidence value for a corresponding entity in the other ontology.

This pre-alignment then goes through semantic validation, which detects semantically invalid mappings and their causes. These invalid mappings are removed from the pre-alignment and logged so that the algorithm does not attempt to map the same entities in a subsequent iteration; mappings are removed from the invalid log when the underlying cause disappears. In the semantic validation process, the pre-alignment is first passed through a pruning process, which detects invalid mappings by analyzing the hierarchical relationships between mapped concepts. This pruning process is performed iteratively until no invalid mappings can be found.

After the pruning process is completed, a graph validation performs a structural analysis using graphs built from the alignment and information from the ontologies, while exploring inconsistencies in equivalence, subsumption, and disjointness relationships. The validation is performed in three phases: class validation, property validation, and concept-property validation. If any invalid mappings are found, the algorithm re-enters the pruning process; otherwise, an alignment is obtained, and the percentage of mappings repeated from the previous alignment is calculated. If this percentage is less than a threshold function, and if the alignment was not previously obtained, the process returns to recalculate the similarity matrices, otherwise the ASMOV system process stops.

Since OAEI 2007, ASMOV has been improved in several important respects. A new, streamlined ontology model has been created, eliminating the use of the Jena ontology model, in order to improve the performance of the system. The lexical similarity calculation has been modified to eliminate the use of Levenshtein distance as an alternative when words are not found in the thesaurus; this calculation, while helping to find some mappings, was also introducing errors, since its value is not comparable to the similarity values obtained using dictionaries. The iterative process has been modified to perform comprehensive pruning and validation in each iteration; this modification has reduced the number of iterations required to find a solution. The ability to use a partial alignment as input to the algorithm has been implemented. A relation classifier has been added to determine whether a relation between two entities mapped to each other is an equality, or whether one is subsumed by the other. And finally, some bugs have been fixed and the overall software code has been improved.

1.3 Adaptations Made for the Evaluation

No special adaptations have been made to the ASMOV system in order to run the 2008 OAEI tests; however, five Java executable classes have been added in order to respectively run the benchmark series of tests, the anatomy tests, the directory tests, the FAO tests, and the conference tests, and output the results in the OAEI alignment format. The threshold function used to determine the stop criteria for ASMOV was established as a step function, 95% for alignments where both ontologies have more than 500 concepts, and 100% otherwise. Although the rules of the contests stated that all alignments should be run from the same set of parameters, it was necessary to change two parameters for the anatomy tests. These parameters relate to the thesaurus being used (UMLS instead of WordNet) and to the flag indicating whether or not to use ids of entities in the lexical similarity calculations.

1.4 Link to the ASMOV System

The ASMOV system (including the parameters file) can be downloaded from http://support.infotechsoft.com/integration/ASMOV/OAEI-2008.

1.5 Link to the Set of Alignments Produced by ASMOV

The results of the 2008 OAEI campaign for the ASMOV system can be found at http://support.infotechsoft.com/integration/ASMOV/OAEI-2008.

2 Results

In this section, we present our comments on the results obtained from the participation of ASMOV in the four tracks of the 2008 Ontology Alignment Evaluation Initiative campaign. All tests were carried out on a PC running SUSE Linux Enterprise Server 10 with two quad-core Intel Xeon processor (1.86 GHz), 8 GB of memory, and 2x4MB cache.

2.1 Benchmark

The OAEI 2008 benchmark tests have been divided by the organizing committee in eleven levels of difficulty; we have added one more level to include the set of 3xx tests, which have been included in the benchmark for compatibility with previous years. The benchmarks for 2008 have varied with respect to 2007 such that the results from both benchmarks are not directly comparable. We have run the OAEI 2008 tests using the current ASMOV implementation and ASMOV from OAEI 2007 [7], which was found to be one of the top three performing systems [8]. The results of these benchmark tests for both versions of ASMOV, as well as the time elapsed for each set of tests, are presented in Table 1.

The precision and recall for the entire suite of tests shows the current implementation of ASMOV achieves 95% precision and 86% recall. This represents a 2% improvement in both precision and recall over the previous version for the entire suite of tests. Moreover, Table 1 shows the significant improvement, of an order of magnitude, in execution time achieved in the 2008 version of ASMOV.

Table 1. Benchmark test results for ASMOV version 2008 and version 2007

Level	ASMOV 2008			ASMOV 2007		
	Precision	Recall	Time (sec)	Precision	Recall	Time (sec)
0	1.00	1.00	8.60	1.00	1.00	103.55
1	1.00	1.00	4.91	1.00	1.00	67.06
2	1.00	0.99	6.06	1.00	1.00	70.11
3	0.98	0.97	9.96	0.99	0.98	143.65
4	0.99	0.98	10.07	1.00	0.96	197.09
5	0.96	0.93	8.14	0.98	0.89	222.43
6	0.94	0.88	7.22	0.92	0.82	203.65
7	0.93	0.83	7.60	0.89	0.77	194.56
8	0.90	0.71	6.65	0.84	0.72	183.82
9	0.78	0.46	2.61	0.70	0.44	79.38
10	0.40	0.04	0.54	0.38	0.05	17.96
3xx	0.81	0.77	3.42	0.82	0.82	130.72
All	**0.95**	**0.86**	**75.78**	**0.93**	**0.84**	**1,613.97**

2.1.1 Levels 0 to 4

ASMOV performs very well in this set of tests, producing an overall precision and recall of close to 100%. In level 3, there is a slight decrease in accuracy, due to test 210, which uses French words for identifiers. We should note that, even if ASMOV

2008 does not use a foreign-language dictionary, it still finds most mappings for test 210, by finding similarities over the hierarchy, property structure, and individual membership of the ontologies. In level 4, the lower precision is skewed due to test 240, where an analysis of the ontologies shows that the two "erroneous" mappings found, `Journal` to `Periodical` and `lastName` to `lastName`, should be considered correct mappings and should be present in the reference alignment.

2.1.2 Levels 5 to 8

In these levels, it can be seen that both the precision and recall diminish as the difficulty level increases, as is expected. It is also clear that there is a significant improvement between our 2007 and 2008 versions in both precision and recall, especially for the higher levels of difficulty. We attribute this improvement to the correction of some bugs in the 2007 version. In general, the tests at these levels have been stripped of labels and/or comments, and have had their ids scrambled, so that lexical similarities are not relevant; ASMOV relies on other ontology features to find a substantial number of correct mappings.

2.1.3 Levels 9 and 10

In levels 9 and 10, the most difficult, there is a pronounced decrease in the precision and recall obtained by ASMOV 2008. The results obtained are nevertheless better than those obtained using the 2007 version. In these tests, the information available in the ontologies useful to make a decision on an alignment is increasingly sparse. Level 10 shows low precision and very low recall results; these are the most difficult tests, where almost no information is available to align the ontologies. In test 262, no mappings were found. In this test, any class could be arbitrarily assigned to any other class, and ASMOV deems that the preferred alignment is the one with no mappings. The other two tests, 265 and 266, have slightly more information in terms of a hierarchy, which permits ASMOV to find some correct mappings.

2.1.4 Test 301-304

As indicated by the organizing committee, these tests represent four real-world ontologies of bibliographic references that contain some imperfections and are included for compatibility with previous years. The overall precision and recall for ASMOV 2008 were respectively 81% and 77%, slightly lower than our 2007 version.

2.2 Anatomy

For the anatomy track, ASMOV uses the UMLS Metathesaurus [5] instead of WordNet in order to more accurately compute the lexical distance between medical concepts. In addition, the lexical similarity calculation between concept names (ids) is ignored as instructed by the track organizers. ASMOV produces an alignment for all four subtasks of this track:

1. *Optimal solution*: The optimal solution alignment is obtained by using the default parameter settings of ASMOV. It took 3 hours and 53 minutes in order to generate an alignment.
2. *Optimal precision*: The alignment with optimal precision is obtained by changing the threshold for valid mappings from 1% to 50%. This means that only mappings with confidences greater or equal to 0.5 make it to the alignment. The time cost for the generation of this alignment was 3 hours and 50 minutes.
3. *Optimal recall*: ASMOV uses a threshold for confidence values of 1%, to avoid negligible non-zero confidences. The alignment with optimal recall is generated by changing this threshold to 0%. Under this setup, it took 5 hours and 54 minutes in order to produce the final alignment.
4. *Extended solution*: The alignment was obtained in 51 minutes. Although one would expect that all the mappings within the partial alignment would make it to the final alignment, ASMOV's semantic validation process rejected two of them. Our analysis of the ontologies justifies the rejection performed by ASMOV.

2.3 Directory

For the 2008 version of ASMOV, we believe that a number of improvements and bug fixes in the semantic validation mechanisms have resulted in a more coherent alignment. A noticeable improvement of ASMOV is in the execution time. It took the 2007 version close to 12 minutes to complete the matching tasks while the current version finished in less than 2 minutes. ASMOV was not used to process the mdirectory tests since it does not yet use a multilingual thesaurus. It also could not run the library and vldr tests due to our inability to run the SKOS-to-OWL converter.

2.4 FAO

ASMOV was able to identify a few mappings in this series of tests. This track helped us refine the ontology modeling component of ASMOV with support for ontology extension through the *owl:imports* construct. The total processing time for the FAO tests was 4 hours and 39 minutes.

2.5 Conference

This collection of tests dealing with conference organization contains 15 ontologies. ASMOV is able to generate 75 generic correspondences from those ontologies. The overall time required to process all 75 correspondences was less than 33 seconds. Manual analysis of a small sample of the alignments produced by ASMOV indicates that the overall output of the classification component is promising.

Some issues were encountered with two of this track’s ontologies: *paperdyne.owl* and *OpenConf.owl*. Specifically, in *paperdyne.owl* the property `hasAcronym` is declared both as a datatype property and as an inverse functional property; in *OpenConf.owl*, an anonymous class is declared as an enumeration of a mixture of classes and individuals. Neither of these constructs is valid in OWL-DL, according to the OWL specification [9]; ASMOV supports only OWL-DL. Additionally, ASMOV had trouble aligning *Conference.owl* and *MICRO.owl*, possibly due to an inability to compare `oneOf` with `Union` concept declarations.

3 General Comments

3.1 Comments on the Results

Although the current version of ASMOV performed well in the 2008 OAEI benchmark series of tests, its accuracy decreased for a subset of the tests compared to the accuracy obtained with last year's version. However the overall precision and recall of the 2008 version of ASMOV performs better than its 2007 counterpart; an improvement of 2% in both precision and recall was attained. Moreover, ASMOV shows a large improvement in its performance and its ability to process larger ontologies, having reduced processing times by one order of magnitude. Nevertheless, further enhancements to its scalability are still needed.

3.2 Discussions on the Way to Improve ASMOV

As in the 2007 version of ASMOV, the mapping validation in the current implementation is still source dependent, making the alignment process a directional one. As our future work, we intend to improve the mapping validation process so that it does not favor the source ontology. Although ASMOV will always converge, the amount of time needed for execution may be too great when dealing with large ontologies. To address this issue a threshold step function was added to the current version of ASMOV. It is necessary to further study different alternatives for a threshold function, in terms of tradeoff between accuracy and scalability.

3.3 Comments on the OAEI 2008 Test Cases

With the new tests added to the benchmark track we were able to do a proper behavior analysis of ASMOV depending on the semantics within ontologies, which guided the correction of coding errors. In the anatomy series of tests, the newly added test, which includes the previously referenced partial alignment, was useful in identifying issues within our semantic validation process; multiple inheritances was not addressed properly and thus led to the rejection of accurate mappings. The directory tests challenged the taxonomy validation of ASMOV while the conference track tested our relation classifier. The FAO tests made sure that ASMOV is able to properly load ontologies that include the *owl:imports* construct.

An ambiguity exists in the instruction of the execution phase of the OAEI 2008 campaign. Participants are told only to use one set of parameters for all tests in all tracks; however, the anatomy track instructs participants to disregard the names (ids) of the concepts and to rely on their labels and the annotation property values in order to perform the lexical comparison. Since the lexical matcher of ASMOV does leverage the id in its computation, a parameter was added to indicate whether or not to use ids. Therefore, the set of parameters for this track was different than for the other ones ASMOV participated in this year. Furthermore, ASMOV uses one of two lexical databases in order to compute the distance between lexical terms. For the anatomy track, the UMLS Metathesaurus was used while WordNet was used for all other tracks.

4 Conclusion

We have provided a brief description of an automated alignment tool named ASMOV, analyzed its performance at the 2008 Ontology Alignment Evaluation Initiative campaign, and compared it with its 2007 version. The test results show that ASMOV is effective in the ontology alignment realm, and because of its flexibility, it performs well in multiple ontology domains such as bibliographic references (benchmark tests) and the biomedical domain (anatomy test). The tests results also showed that with improvement in execution time, ASMOV is now a practical tool for real-world applications that require on-the-fly alignments of small ontologies.

Acknowledgments. This work is funded by the National Institutes of Health (NIH) under grant R43RR018667. The authors also wish to acknowledge the contribution of Mr. Patrick Shironoshita of INFOTECH Soft, Inc.

References

1. Euzenat J and Shvaiko P. Ontology Matching. Springer-Verlag, Berlin Heidelberg, 2007.
2. Euzenat J. and Valtchev P. Similarity-based ontology alignment in OWL-lite. *In Proc. 15th ECAI*, Valencia (ES), 2004, 333-337.
3. Jean-Mary Y., Kabuka, M. ASMOV: Ontology Alignment with Semantic Validation. Joint SWDB-ODBIS Workshop, September 2007, Vienna, Austria, 15-20
4. Jena from HP Labs http://www.hpl.hp.com/semweb/
5. Unified Medical Language System (UMLS) http://umlsks.nlm.nih.gov/
6. WordNet http://wordnet.princeton.edu/
7. Jean-Mary Y, Kabuka M. ASMOV: Results for OAEI 2007. http://www.dit.unitn.it/~p2p/OM-2007/3-o-ASMOV_OAEI_2007.pdf. Accessed 24 Sept 2008.
8. Euzenat J, et.al. Results of the Ontology Alignment Evaluation Initiative 2007. http://www.dit.unitn.it/~p2p/OM-2007/0-o-oaei2007.pdf. Accessed 24 Sept 2008.
9. Mike Dean and Guus Schreiber, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/owl-ref/

Ontology Matching with CIDER: Evaluation Report for the OAEI 2008

Jorge Gracia, Eduardo Mena

IIS Department, University of Zaragoza, Spain
{jogracia,emena}@unizar.es

Abstract. Ontology matching, the task of determining relations that hold among terms of two different ontologies, is a key issue in the Semantic Web and other related fields. In order to compare the behaviour of different ontology matching systems, the Ontology Alignment Evaluation Initiative (OAEI) has established a periodical controlled evaluation that comes in a yearly event. We present here our participation in the 2008 initiative.

Our schema-based alignment algorithm compares each pair of ontology terms by, firstly, extracting their ontological contexts up to a certain depth (enriched by using transitive entailment) and, secondly, combining different elementary ontology matching techniques (e.g., lexical distances and vector space modelling). Benchmark results show a very good behaviour in terms of precision, while preserving an acceptable recall.

Based on our experience, we have also included some remarks about the nature of benchmark test cases that, in our opinion, could help improving the OAEI tests in the future.

1 Presentation of the system

In [7] we presented a system that analyzes a keyword-based user query, in order to automatically extract and make explicit, without ambiguities, its semantics. Firstly, it discovers and extracts candidate senses (expressed as ontology terms) for each keyword, by harvesting the Semantic Web. Local ontologies or lexical resources, as WordNet [6], can also be accessed. Then, an alignment and integration step is carried out in order to reduce redundancies (many terms from different ontologies could describe the same intended meaning, so we integrate them as a single sense). Finally, a disambiguation process is run to pick up the most probable sense for each keyword, according to the context. The result can be eventually used in the construction of a well-defined semantic query (expressed in a formal language) to make explicit the intended meaning of the user.

We realized that the alignment component of our system is general enough to be used for many other tasks so, based on it, we have developed an independent aligner to be evaluated in the OAEI contest[1]. The latest version of our alignment

[1] http://oaei.ontologymatching.org/2008/

service is called CIDER (Context and Inference baseD alignER), which is the subject of this study. It relies on a modified version of the semantic similarity measure described in [7].

1.1 State, purpose, general statement

According to the high level classification given in [3], our method is a *schema-based* system (opposite to others which are instance-based, or mixed), because it relies mostly on schema-level input information for performing ontology matching. As it was mentioned in the previous section, the initial purpose of our algorithm was to discover similarities among possible senses of user keywords, in order to integrate them when they were similar enough (to be later disambiguated and used in semantic query construction). Therefore, our alignment algorithm was initially applied to a previously discovered set of ontological terms, describing possible senses of a keyword.

For this study we have generalized the method, to admit any two ontologies, and a threshold value, as input. Comparisons among all pairs of ontology terms (not only the ones that could refer to a same user keyword) are established, producing as output an RDF document with the obtained alignments.

1.2 Specific techniques used

Our alignment process takes as basis a modified version of the *semantic similarity measure* described in [7]. A detailed discussion of the introduced improvements is out of the scope of this paper. However, here is a brief summary of them:

1. Addition of a *transitive entailment* mechanism during the extraction step, which has remarkably improved our results in terms of quality.
2. Enrichment of our initially naive comparisons between instances, by considering also their properties and corresponding values.
3. Optimization of the initially costly comparisons among properties of concepts, substituting their recursive focus with the use of vector space modelling. We have found that it preserves quality, while reduces time significantly.

Figure 1 shows a schematic view of the way our matcher works. O_1 and O_2 represent the input ontologies. M is the matrix of resultant comparisons among ontology terms, and A is the extracted alignment.

The first step is to extract the ontological context of each involved term, up to a certain depth. That is (depending on the type of term), their synonyms, textual descriptions, hypernyms, hyponyms, properties, domains, roles, associated concepts, etc. This process is enriched by applying a transitive inference mechanism, in order to add more semantic information that is not explicit in the asserted ontologies.

The second step is the computation of similarity for each pair of terms. It is carried out differently, depending on the type of ontology term (concept, property or individual). Without entering into details, comparisons are performed like this:

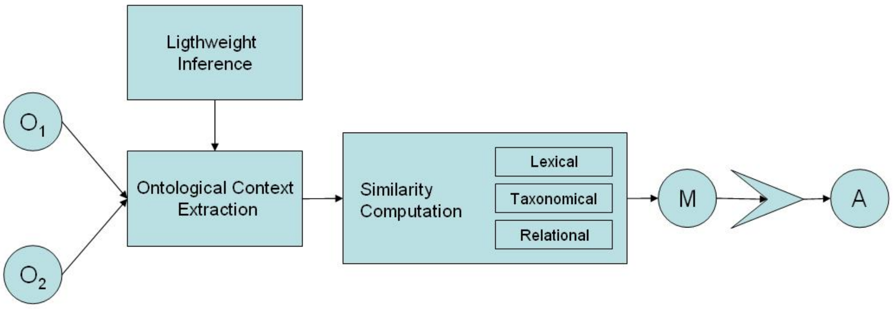


Fig. 1. Scheme of the CIDER process.

1. Linguistic similarity between terms, considering labels and descriptions, is computed.
2. A subsequent computation explores the structural similarity of the terms, exploiting their ontological contexts and using vector space modelling in comparisons. It comprises comparison of taxonomies and relationships among terms (e.g. properties of concepts).
3. The different contributions are weighted, and a final similarity degree is provided.

After that, a matrix M with all similarities is obtained. The final alignment A is then extracted, finding the highest rated one-to-one relationships among terms, and filtering out the ones that are below the given threshold.

In terms of implementation, CIDER prototype has been developed in Java, extending the Alignment API [2]. The input are ontologies expressed in OWL or RDF, and the output is served as a file expressed in the *alignment format* [2], although it can be easily translated to other formats as well.

1.3 Adaptations made for the evaluation

As the benchmark test does not consider mappings between instances, we have not computed instance alignment for this particular test. No other adaptations have been needed.

1.4 Link to the system and parameters file

The version of CIDER used for this evaluation can be found at
http://sid.cps.unizar.es/SEMANTICWEB/ALIGNMENT/OAEI08/

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

The obtained alignments for the contest can be found at
http://sid.cps.unizar.es/SEMANTICWEB/ALIGNMENT/OAEI08/results/CIDER.zip

2 Results

The following subsections describe the participation of our system in two tracks of the contest: benchmark and directory. Some remarks specific to each test are described, as well as a tentative explanation of the obtained results. Further information about the whole results of the contest can be found at [1].

2.1 Benchmark

The target of this experiment is the alignment of bibliographic ontologies. A reference ontology is proposed, and many comparisons with other ontologies of the same domain are performed. The tests are systematically generated, modifying differently the reference ontology in order to evaluate how the algorithm behaves when the aligned ontologies differ in some particular aspects. A total of 111 test cases have to be evaluated. They are grouped in three sets:

1. Concept test (cases *1xx*: 101, 102, ...), that explore comparisons between the reference ontology and itself, described with different expressivity levels.
2. Systematic (cases *2xx*). It alters systematically the reference ontology to compare different modifications or different missing information.
3. Real ontology (cases *3xx*), where comparisons with other "real world" bibliographic ontologies are explored.

We cannot provide results for benchmark cases 202 and 248-266, because our system does not deal with ontologies in which syntax is not significant at all (these cases present a total absence or randomization of labels and comments). Consequently, we expect a result with a low recall in this experiment, as the benchmark test unfavours methods that are not based on graph structure analysis (or similar techniques).

In Table 1 we show the obtained results, grouped by type of cases. We have obtained a very high precision (97%), which is in the top-three best values obtained in the contest (out of 13 participants), while recall has been lower (62%), due to the above mentioned reason. The extended results for the complete dataset has been published separately by the organizers[2].

	1xx	**2xx**	**3xx**	**Average**	**H-Mean**
Precision	0.99	0.97	0.90	0.97	0.97
Recall	0.99	0.60	0.73	0.61	0.62

Table 1. Averaged results for the benchmark dataset.

Alternatively to the official results, we have computed the precision and recall of the benchmark test excluding the cases 202 and 248-266 (and their variations

[2] http://oaei.ontologymatching.org/2008/results/benchmarks.html

248-2, 248-4, etc.), in which ontology terms are described with non expressive texts. This is an "internal" exercise, which does not let us direct comparisons with other methods in the contest, but gives us another point of view (more accurate, according to the final usage of our system) of the behaviour of our method. Results are given in Table 2.

	1xx	2xx	3xx	Average	H-Mean
Precision	0.99	0.97	0.90	0.96	0.97
Recall	0.99	0.87	0.73	0.87	0.86

Table 2. Results for the benchmark dataset omitting cases with no significant texts.

2.2 Directory

The objective of this experiment is to match terms from plain hierarchies, extracted from web directories. It consist of more than 4 thousand elementary alignments. We consider that our method cannot show all its strengths in this, because the available information is extremely sparse, lacking in semantic descriptions beyond hierarchical relationships (no instances, no properties, no comments, no synonyms, ...).

Results have been: 60% precision, 38% recall and 47% F-measure, which has been the second best result in this year competition (out of seven participants). A detailed comparison has been published by organizers[3]. We see that, even directory alignment is not the target of our system, it behaves reasonably well when matching plain hierarchies.

3 General comments

The following subsections contain some remarks and comments about the results obtained, as well as about the test cases and evaluation process.

3.1 Comments on the results

As expected, we obtained better precision than recall in the benchmark test (due to the reasons mentioned in Section 2.1). Also in the directory experiment precision was higher than recall. However, it is consistent with the fact that our alignment is targeted to be used in an automatic way, minimizing human intervention. In this conditions, precision have to be promoted over recall. That is, maybe our system does not discover all correspondences, but we have to be sure that, in case it discovers an equivalence mapping between two terms, they

[3] http://www.disi.unitn.it/∼pane/OAEI/2008/directory/result/

are most likely referring to the same meaning. Otherwise their later integration would be erroneous, and the mistake would eventually be propagated to the other steps of the system.

3.2 Discussions on the way to improve the proposed system

Our method does not consider extensional information when comparing concepts, focusing only on the semantic description of the terms in the corresponding ontologies. Its inclusion could improve results in some cases where this information is available.

Additionally, although our system considers many features of ontologies, their richness vary a lot from one case to another. We consider that the addition of mechanisms to auto-adjust weights to the characteristics of ontologies (as they do in [5]) could largely benefit our method.

Finally, although our similarity measure has been much optimized, in terms of time response, the overall alignment process can still be subject of further improvement.

3.3 Comments on the OAEI 2008 test cases

We have found the benchmark test very useful as a guideline for our internal improvements of the method, as well as to establish a certain degree of comparisons with other existing methods.

On the other hand, we have missed some important issues that are not taken into account in the systematic benchmark series:

1. Benchmark tests only consider positive matchings, not measuring the ability of different methods to avoid links among barely related ontologies (only case 102 of benchmark goes in that direction).
2. For our purposes, we try to emulate the human behaviour when mapping ontological terms. As human experts cannot properly identify mappings between ontologies with scrambled texts, neither does our system. However, reference alignments provided in the benchmark evaluation for cases 202 and 248-266, do not follow this intuition. We hope this bias will be reduced in future contests.
3. Related to the latter, cases in which equal topologies, but containing different semantics, lead to false positives, are not explicitly taken into account in the benchmark.
4. How ambiguities can affect the method is not considered either in the test cases. It is a consequence of using ontologies belonging to the same domain. For example, it would be interesting to evaluate how “film” in an ontology about movies, is mapped to “film” as a “thin layer” in another ontology. Therefore it is difficult to evaluate the benefits of including certain disambiguation techniques in ontology matching [4].

3.4 Comments on the OAEI 2008 measures

Unsuitability of precision and recall measures for ontology matching evaluation is a well known problem [3]. We encourage organizers to try different measures that count all correct found correspondences, even when they are not explicit in the reference alignment.

4 Conclusion

We have presented here some results of our first participation in the OAEI 2008 contest. We have limited to two tracks: benchmark and directory, but we hope to extend our participation in the future.

Our schema-based alignment algorithm compares the ontological contexts of each pair of terms (enriched with transitive inference) by combining different elementary ontology matching techniques (comparing vocabulary, taxonomies, relations,...). Benchmark results show a very good behaviour of our system in terms of precision, while keeping an acceptable recall. It confirms the validity of the measure we have conceived, and its suitability to be applied in ontology matching tasks. It encourages us to tackle further improvements, and to extend the scope and applicability of our techniques.

We have also included, based on our experience, some considerations about the nature of benchmark test cases that, in our opinion, could help improving future contests.

Acknowledgments. This work is supported by the CICYT project TIN2007-68091-C02-02.

References

1. C. Caracciolo, J. Euzenat, L. Hollink, R. Ichise, A. Isaac, V. Malaisé, C. Meilicke, J. Pane, P. Shvaiko, H. Stuckenschmidt, O. Šváb-Zamaza, and V. Svátek. First results of the ontology alignment evaluation initiative 2008. In *In Proc. ISWC-2008 Workshop on Ontology Matching*, 2008.
2. J. Euzenat. An API for ontology alignment. In *3rd International Semantic Web Conference (ISWC'04), Hiroshima (Japan)*. Springer, November 2004.
3. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, 2007.
4. J. Gracia, V. López, M. d'Aquin, M. Sabou, E. Motta, and E. Mena. Solving semantic ambiguity to improve semantic web based ontology matching. In *Proc. of 2nd Ontology Matching Workshop (OM'07), at 6th International Semantic Web Conference (ISWC'07), Busan (Korea)*, November 2007.
5. Y. Jean-Mary and M. Kabuka. ASMOV: Ontology alignment with semantic validation. In *Proc. of Joint SWDB-ODBIS Workshop on Semantics, Ontologies, Databases, Vienna (Austria)*, September 2007.
6. G. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), nov 1995.
7. R. Trillo, J. Gracia, M. Espinoza, and E. Mena. Discovering the semantics of user keywords. *Journal on Universal Computer Science. Special Issue: Ontologies and their Applications*, November 2007.

DSSim Results for OAEI 2008

Miklos Nagy[1], Maria Vargas-Vera[1], and Piotr Stolarski[2]

[1] The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
mn2336@student.open.ac.uk;m.vargas-vera@open.ac.uk
[2] Poznan University of Economics
al. Niepodleglosci 10, 60-967 Poznan, Poland
P.Stolarski@kie.ae.poznan.pl

Abstract. The growing importance of ontology mapping on the Semantic Web has highlighted the need to manage the uncertain nature of interpreting semantic meta data represented by heterogeneous ontologies. Considering this uncertainty one can potentially improve the ontology mapping precision, which can lead to better acceptance of systems that operate in this environment. Further the application of different techniques like computational linguistics or belief conflict resolution that can contribute the development of better mapping algorithms are required in order to process the incomplete and inconsistent information used and produced during any mapping algorithm. In this paper we introduce our algorithm called "DSSim" and describe the improvements that we have made compared to OAEI 2006 and OAEI 2007.

1 Presentation of the system

1.1 State, purpose, general statement

Ontology mapping systems need to interpret heterogeneous data in order to simulate "machine intelligence", which is a driving force behind the Semantic Web. This implies that computer programs can achieve a certain degree of understanding of such data and use it to reason about a user specific task like question answering or data integration. In practice there are several roadblocks[1] that hamper the development of mapping solutions that perform equally well for different domains. Additionally the different combination of these challenges needs to be addressed in order to design systems that provides good quality results. DSSim has been designed to address the combination of the 3 following challenges:

- Representation and interpretation problems: Ontology designers have a wide variety of languages and language variants to choose from in order to represent their domain knowledge. The most widely used for small and medium sized ontologies are RDF(S) and OWL as Web ontology language however OWL has three increasingly-expressive sublanguages(OWL Lite, OWL DL, OWL Full) with different expressiveness and language constructs. Other languages like SKOS, which is a standard to support the use of knowledge organization systems (KOS) such as thesauri, classification schemes, subject heading systems and large scale taxonomies within the framework of the Semantic Web. From the logical representation point of view each representations are valid separately and no logical reasoner would find inconsistency in them individually. However the problem occurs once we need to compare ontologies with different representations in order to determine the similarities between classes and individuals. Consider for example one ontology where the labels are described with standard class *rdfs:label* tag and an another ontology where the same is described as *hasNameScientific* data property. As a result of these representation differences ontology mapping systems will always need to consider the uncertain aspects of how the semantic web data can be interpreted.

- Quality of the Semantic Web data: For every organisation or individual the context of the data, which is published can be slightly different depending on how they want to use their data. Therefore from the exchange point of view incompleteness of a particular data is quite common. The problem is that fragmented data environments like the Semantic Web inevitably lead to data and information quality problems causing the applications that process this data deal with ill-defined inaccurate or inconsistent information on the domain. The incomplete data can mean different things to data consumer and data producer in a given application scenario. In traditional integration scenarios resolving these data quality issues represents a vast amount of time and resources for human experts before any integration can take place. The main problem what Semantic Web applications need to solve is how to resolve semantic data quality problems i.e. what is useful and meaningful because it would require more direct input from the users or creators of the ontologies. Clearly considering any kind of designer support in the Semantic Web environment is unrealistic therefore applications itself need to have built in mechanisms to decide and reason about whether the data is accurate, usable and useful in essence, whether it will deliver good information and function well for the required purpose.

- Efficient mapping with large scale ontologies: Ontologies can get quite complex and very large, causing difficulties in using them for any application. This is especially true for ontology mapping where overcoming scalability issues becomes one of the decisive factors for determining the usefulness of a system. Nowadays with the rapid development of ontology applications, domain ontologies can became very large in scale. This can partly be contributed to the fact that a number of general knowledge bases or lexical databases have been and will be transformed into ontologies in order to support more applications on the Semantic Web. This year the OAEI tracks have also included a task very large cross lingual ontologies, which includes establishing mappings between Wordnet, DBPedia an GTAA(Dutch acronym for Common Thesaurus for Audiovisual Archives), which is a domain specific thesaurus with approximately 160.000 terms. A lot of researcher might argue that the Semantic Web is not just about large ontologies created by the large organisations but more about individuals or domain experts who can create their own relatively small ontologies and publish it on the Web. Indeed might be true however from the scalability point of view it does not change anything if thousands of small ontologies or a number of huge ontologies need to be processed.

As a result from the mapping point of view ontologies will always contain inconsistencies, missing or overlapping elements and different conceptualisation of the same terms, which introduces a considerable amount of uncertainty into the mapping process. In order to represent and reason with this uncertainty authors (Vargas-Vera and Nagy) have proposed a multi agent ontology mapping framework [2], which uses the Dempster-Shafer [3] theory in the context of Question Answering. Since our first proposition[4] of such solution in 2005 we have gradually developed and investigated multiple components of such system and participated in the OAEI in order to validate the feasibility of our proposed solution. Fortunately during the recent years our original concept has received attention from other researchers [5, 6], which helps to broaden the general knowledge on this area. We have investigated different aspects of our original idea namely the feasibility of belief combination[7] and the resolution of conflicting beliefs [8] over the belief in the correctness of similarities using the fuzzy voting model. A comprehensive description of the Fuzzy voting model can be found [8]. For this contest (OAEI 2008) the benchmarks, anatomy, fao, directory, mldirectory, library and vlcr tracks had been tested with this new version of DSSim (v0.3). Therefore, we had improved our precision and recall measures. Furthermore, experiments(based on the benchmarks) reported in [8] showed that average recall can be improved up to 12% and average recall up to 16%. These new improvements have been included into our DSSim v0.3 system and been tested through OM-2008.

1.2 Specific techniques used

This year we introduced also two types of improvements. Those enhancements are mainly connected to multiword ontology entity labels and include: compound nouns comparisons with the use of semantic relations technique as well as extensive production of abbreviations based on defined language rules. The realization of the first improvement comes from the inspiration of researches on computational linguistics, whereas the second advancement is produced on the basis of pragmatic observations of exemplary unmatched alignments from the conference track.

A fundamental case which has led us to the idea of introducing the abbreviations factory - a component responsible for production of expected possible shortenings of words or phrases - came from the Conference track. The available linguistic resources (i.e. Wordnet) provide indeed a very extensive aid in dealing with different sorts of language processing tasks. Nevertheless, those resources are not ideal. As a result the mentioned Wordnet, for instance, does not offer any service for obtaining any list of shortened forms for a word or phrase. Though it may seem less important in the task of ontology matching we may consider a straight-ahead example invalidating such a view.

In some conference-track ontologies there were entities (mostly classes) denoting the concept of “Program Committee” (or “Program Committee Member”). Of course any human being with a little acquaintance of the domain would know that the phrase is commonly abbreviated to “PC” (often encountered in the ontologies). Unfortunately, such knowledge comes rather from the experience and cannot be expected to be ad-hoc part of a computer system. Another important observation is that some specific abbreviations are typical only for those specific domains and can reflect even completely other phrases in a common language. For instance the "PC" phrase would rather be interpreted as a shortening for "Personal Computer". In fact only few on-line abbreviation dictionaries return the sense of "Program Committee", which hindered us from using the external resource on the favor of trying a (simpler for implementation) rule-based shortenings generator.

The compound nouns comparison method is an interesting example of algorithm dealing with interpretation of compound nouns based on earlier works done in such fields as language understanding as well as question-answering and machine translation. The problem of establishing the semantic relations between items of compound nouns has awaited many different approaches [9] [10] [11]. Yet, all of them should be regarded as partial solutions rather than a definite one. Most of the cases uses either manually created rules [9] or machine learning techniques [10] in order to automatically build classification rules that will enable to rate any given compound noun phrase into one of a set of pre-selected semantic relations which best reflects the sense and nature of that phrase. As mentioned, most approaches are not comprehensive and their authors limit their resolutions to some specific restrictions. For instance the most often case that is being scrutinized is the binary type of compound nouns, where the compound phrase is made up of only two nouns (a head and modifier).

In the context of ontology matching, the class of compound nouns semantic relation detection algorithms may be used in order to determine such relations within ontology entities' identifiers and labels. After the relation $r^{1,n}$ has been classified independently for entities in the first of aligned ontologies O^1 and $r^{2,m}$ separately for entities form the other ontology O^2, the alignments may be produced between the entities from O^1 and O^2 on the basis of similarity between the relations $r^{1,n}$ and $r^{2,m}$ itself.
Such approach has its disadvantages but those can be in large part eliminated by introducing the algorithm into more general matching framework. For instance it fits especially well into the aligning system implemented by DSSim (described in details in [2]). As the number of elements in the set of isolated semantic relations is usually limited only to very general ones, the probability of detecting the same or similar relations is subjectively high, therefore the method itself is rather sensitive to the size of the set. Yet even if that number is relatively small the method may still be helpful if the outcomes are combined with other ways of similarity assessment. In the case of DSSim it means that the method can be treated as one of the experts.

Our implementation is, so far, a vastly simplified one. In the research we initially propose a small set of manually created rules, which employ different entries describing the ontology entity (comments are most favourable). This means that for our purposes we adopted some parts of method given in [9] but the way the rules are created rather moves our approach next to [10]. The method delimits us to processing of binary compounds, but it is potentially possible to change this. We started the experiments with a general set of semantic relations but recognized the need to switch to domain specific - depending on the type and field of representation of ontologies. Such creation of relations' set is another challenge. We also use only two-state logic in expressing similarity function between the classified semantic relations.

We will consider some simple examples describing in more details the practical aspects of our implementation. Table 1 represent exemplary general relations that we

Relation type	Rule
CAUSE	“make”
EFFECT	“result|effect”
LOCATION_FROM	“originates at|starts from”
AGENT	“perform|performing”
INSTRUMENT	“use|employ”
POSSESSOR	“has|have|possess|own|owns”
PRODUCT	“produce|create”
EQUATIVE	“is also|equal”
PROPERTY	“is”

Table 1. A model way of defining compound nouns semantic relations and classification rules.

defined (on the basis of our analysis and cited paper [9]). The created rules are simple using the formalism of regular expressions and can mainly be based on the recognition of keywords in the additional descriptions (comments) of the compound phrases. Some example keywords are also presented in the table.

Now let us take into account the case of earlier mentioned “Program Committee Member”. Assuming that the ontology entity of this name is also accompanied with the comment: “[...] person that is performing the tasks on the account of the Program Committee [...]” will result in triggering the conclusion that this is the instance of the AGENT type of semantic relation once the rules are run against the comment[3].

If during processing of the other ontology in the matching task another entity of the same meta-class (i.e. a concept, relation, etc.) will be detected and it will also be assigned the same semantic relation category (AGENT) then the binary nature of the comparison will deem those entities to be equal. It should be noted that so far any other relation within our implementation cannot be considered ”similar”. Thus, entities with other relations like EFFECT or POSSESOR are simply regarded as not the same. Nevertheless as stressed above, it is vital to take into account that having two compound nouns entities recognized with the same category does not mean that those entities really refer to the same meaning. So such fact should be viewed rather as a premise, at least in the case of general semantic relations categories[4].

Summing up the introduced improvements, the compound nouns comparisons with the use of semantic relations technique is a very promising method that should in the future seriously improve the outcome of our algorithm. Nevertheless, as it is still in a rather premature phase the results are not according to expectations, yet. The method is however under severe development and thus we expect major impact on results in the future versions. It would also be interesting to introduce machine-learning techniques for rules acquisition instead of use only manually implemented ones - for instance Artificial Neural Networks can be used for this task.

[3] Because of the existence of the “performing” keyword.

[4] The level of certainty is higher when the ontologies are connected to a narrower domain or if the more specific categories are introduced.

1.3 Adaptations made for the evaluation

Our ontology mapping system is based on a multi agent architecture where each agent built up a belief for the correctness of a particular mapping hypothesis. Their beliefs are then combined into a more coherent view in order to provide better mappings. Although for the previous OAEI contests we have re-implemented our similarity algorithm as a standalone mapping process which integrates with the alignment api, we have recognised the need for possible parallel processing for tracks which contain large ontologies e.g. very large cross-lingual resources track. This need is indeed coincide with our original idea of using distributed multi-agent architecture, which is required for scalability purposes once the size of the ontology is increasing. Our modified mapping process can utilise multi core processors by splitting up the large ontologies into smaller fragments. Both the fragment size and the number of cores that should be used for processing can be set in the "param.xml" file.

Based on the previous implementation we have modified our process for the OAEI 2008 which works as follows:

1. Based on the initial parameters divide the large ontologies into n*m fragments.
2. Parse the ontology fragments and submit them into the alignment job queue.
3. Run the job scheduler as long as we have jobs n the queue and assign jobs into idle processor cores.
   - 3.1 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. Our algorithm consults WordNet in order to augment the query concepts and properties with their hypernyms.
   - 3.2 We take syntactically similar concepts and properties to the query graph from ontology 2 and build a local ontology graph that contains both concepts and properties together with the close context of the local ontology fragments.
   - 3.3 Different similarity and semantic 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 subjective assessment.
   - 3.4 Then the similarity matrixes are used to determine belief mass functions which are combined using the Dempster's rule of combination. Based on the combined evidences we select those mappings in which we calculate the highest belief function.
4. The selected mappings are added into the alignment.

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

1.4 Link to the system and parameters file

http://kmi.open.ac.uk/people/miklos/OAEI2008/tools/DSSim.zip

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

http://kmi.open.ac.uk/people/miklos/OAEI2008/results/DSSim.zip

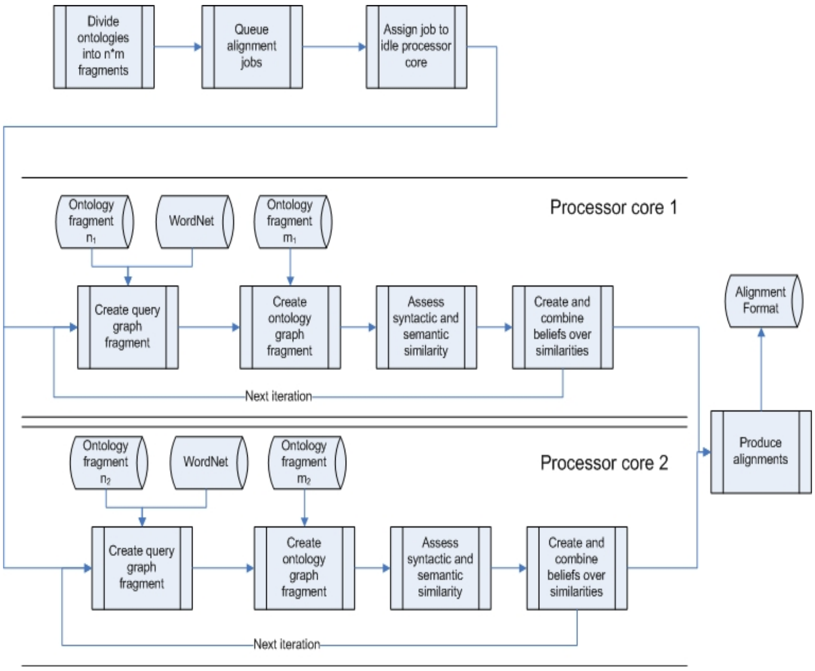

Divide ontologies into n*m fragments
Queue alignment jobs
Assign job to idle processor core
Processor core 1
Ontology fragment n_1
WordNet
Ontology fragment m_1
Create query graph fragment
Create ontology graph fragment
Assess syntactic and semantic similarity
Create and combine beliefs over similarities
Next iteration
Alignment Format
Produce alignments
Processor core 2
Ontology fragment n_2
WordNet
Ontology fragment m_2
Create query graph fragment
Create ontology graph fragment
Assess syntactic and semantic similarity
Create and combine beliefs over similarities
Next iteration

Fig. 1. The mapping process on a dual-core processor

2 Results

2.1 benchmark

The benchmarks have been extended with new tests this year, which allows a more fine-grained evaluation of the results. It is definitely more difficult than last contest (2007). However our algorithm has produced the same results as last year. If we do not consider the new tests we have improved the recall with keeping the same precision. The weakness of our system to provide good mappings when only semantic similarity can be exploited is the direct consequence of our mapping architecture. At the moment we are using four mapping agents where 3 carries our syntactic similarity comparisons and only 1 is specialised in semantics. However it is worth to note that our approach seems to be stable compared to our last years performance, as our precision recall values were similar in spite of the fact that more and more difficult tests have been introduced in this year. As our architecture is easily expandable with adding more mapping agents it is possible to enhance our semantic mapping performance in the future. The overall conclusion is that our system produces stable quality mappings, which is good however we still see room for improvements.

2.2 anatomy

The anatomy track contains two reasonable sized real world ontologies. Both the Adult Mouse Anatomy (2.744 classes) and the NCI Thesaurus (3.304 classes) describes anatomical concepts. The classes are represented with standard owl:Class tags with proper rdfs:label tags. Our mapping algorithm has used the labels to establish syntactic similarity and has used the rdfs:subClassOf tags to establish semantic similarities between class hierarchies. We could not make use of the owl:Restriction and oboInOwl: hasRelatedSynonym tags as this would require ontology specific additions. The anatomy track represented a number of challenges for our system. Firstly the real word medical ontologies contain classes like “outer renal medulla peritubular capillary”, which cannot be easily interpreted without domain specific background knowledge. Secondly one ontology describes humans and the second describes mice. To find semantically correct mappings between them requires deep understanding of the domain. According to the results our system DSSim did not perform as well as we have expected in this test compared to the best system (SAMBO) because we do not use any domain specific background knowledge or heuristics but the standard WordNet dictionary. The run time per test was around 30 min, which is an improvement compared to last year.

2.3 fao

The fao track contains one reasonable sized and two large ontologies. The AGROVOC describes the terminology of all subject fields in agriculture, forestry, fisheries, food and related domains (e.g. environment). It contains around 2.500 classes. The classes itself are described with a numerical identifier through rdf:ID attributes. Each class has an instance, which holds labels in multiple languages describing the class. For establishing syntactic similarity we substitute the class label with its instance labels.

Each instance contains a number of additional information like aos:hasLexicalization of aos:hasTranslation but we do not make use of it as it describes domain specific information. ASFA contains 10.000 classes and it covers the world's literature on the science, technology, management, and conservation of marine, brackish water, and freshwater resources and environments, including their socio-economic and legal aspects. It contains only classes and its labels described by the standard owl:Class formalism. The fisheries ontology covers the fishery domain and it contains a small number of classes and properties with around 12.000 instances. Its conceptual structure is different from the other two ontologies. These differences represented the major challenge for creating the alignments. The FAO track was one of the most challenging ones as it contains three different sub tasks and large scale ontologies. As a result DSSim was one of the two systems, which could create complete mappings. The other systems have participated in only one sub task. In terms of overall F-Value RiMOM has performed better than DSSim. This can be contributed to the fact that the FAO ontologies contain all relevant information e.g. *rdfs:label*, *hasSynonym*, *hasLexicalisation* on the individual level and using them would imply implementing domain specific knowledge into our system. Our system has underperformed RiMOM because our individual mapping component is only part of our whole mapping strategy whereas RiMOM could choose the favour instance mapping over other strategies. However in the agrorgbio sub task DSSim outperformed RiMOM, which shows that our overall approach is comparable. The total execution time was around 10 hours.

2.4 directory

In the library track only 6 systems have participated this year. In terms of F-value DSSim has performed the best however the difference is marginal compared to the CIDER or Lily systems. The directory test as well has been manageable in terms of execution time. In general the large number of small-scale ontologies made it possible to verify some mappings for some cases. The tests contain only classes without any labels but in some cases different classes have been combined into one class e.g. "News_and_Media" that introduces certain level of complexity for determining synonyms using any background knowledge. To address these difficulties we have used a compound noun algorithms described in section 1.2. The execution time was around 15 minutes.

2.5 mldirectory

This track contains ontologies from five domains namely automobile, movie, outdoor, photo and software in both English and Japanese. They contain class descriptions in OWL format and RDF descriptions for the instances with labels and comments. We have produced only the English-English class alignments using the instance labels for the classes where possible. There were no reference alignments for this track and no expert evaluation were carried out for the results. The evaluators have compared the systems based on how many alignments the systems produced and what was overlap between the provided results. In the english-english alignment only 4 systems have participated from which only three (including DSSim) has run all data sets. Based on the total alignment provided DSSim achieved the third place(around 30 % less mappings compared to RiMOM but only 5% less than Lily). The most surprising concerning the results is that there were no mappings, which were provided by all 4 systems at any of the data sets. From the performance point of view the run time for this track was around 8 hours.

2.6 library

The library track contains two SKOS describing scientific collections (GTT) which is a huge vocabulary containing 35.000 general concepts and the Brinkman thesaurus, containing a large set of headings with more than 5.000 descriptions. Additionally not all labels were available in English therefore we have used the original Dutch labels. The implication is that we could not determine hypernyms from WordNet, which might impact our mapping precision negatively. We have participated in this track last year as well and although we did not add Dutch background knowledge to our algorithm we expect improvements compared to last year. The track is difficult partly because of its relative large size and because of its multilingual representation. Nevertheless in the library track DSSim has performed the best out of the 3 participating systems. The track is difficult partly because of its relative large size and because of its multilingual representation. However these ontologies contain related and broader terms therefore the mapping can be carried out without consulting multi lingual background knowledge. This year the organisers have provided instances as separate ontology as well however we did not make use of it for creating our final mappings. For further improvements in recall and precision we will need to consider these additional instances in the future. This year the run time was around 12 hours.

2.7 vlcr

This vlcr track was the most complex this year and DSSim was the only system that have participated in this track. It contains 3 large ontologies. The GTAA thesaurus is a Dutch public audiovisual broadcasts archive, for indexing their documents, contains around 3.800 subject keywords, 97.000 persons, 27.000 names and 14.000 locations. The DBPedia is an extremely rich dataset. It contains 2.18 million resources or ”things”, each tied to an article in the English language Wikipedia. The ”things” are described by titles and abstracts in English and often also in Dutch. We have converted the original format into standard SKOS in order to use it in our system. However we have converted only the labels in English and in Dutch whenever it was available. The third resource was the WordNet 2.0 in SKOS format where the synsets are instances rather than classes. In our system the WordNet 3.0 is included into as background knowledge therefore we have converted the original noun-synsets into a standard SKOS format and used our WordNet 3.0 as background knowledge. In this track our precision has ranged from 10% to 94% depending on the test and facet. The lowest precision 0.1 occurred on the GTAA-Wordnet mapping for the persons facet. This can be explained because the GTAA contains nearly hundred thousand persons, which does not have at all correspondence in WordNet. In fact WordNet contains very few persons. As the number of entities in these ontologies are very large only an estimation was can be calculated for the recall/coverage and for not all the facets. The estimated recall values for the evaluated samples were relatively low around 20 %. For more advanced evaluation more tests will be needed in order to identify the strengths and weaknesses of our system. The run time of the track was over 2 weeks and we could not run the complete GTAA-DBPedia combination due to the lack of time.

2.8 conferences

This test set is made up of collection of 15 real-case ontologies dealing with the domain of conference organization. Although all the ontologies are well embedded in the described field, nevertheless they are heterogeneous in their nature. This heterogeneity comes mainly from: designed ontology application type, ontology expressivity in terms of formalism, and robustness. Out of given 15 ontologies the production of alignments should result in 210 possible combinations (we treat the equivalent alignment as symmetric). However, we obtained 91 non-empty alignment files in the generation. DSSim was one of two participants, which provided the maximum 105 alignments this year. The results were evaluated based on different methods e.g. sample and approximate or reference alignments. Fortunately this year a new reference alignment has been produced, which contains all possible pairs of five ontologies. Three confidence threshold was used for the evaluation(0.2, 0.5, 0.7) where the given threshold was used to filter results with the given threshold. Based on the F-measure our system performed differently considering the given threshold values. With threshold 0.2 DSSim is on the third position out of the three participating systems where the difference between systems is marginal. The position changes as the threshold increases. Using 0.5 as threshold DSSim moves to the first position while maintaining a marginal difference compared to the second place. The situation changes considerably using the 0.7 threshold. The difference between DSSim and the other systems increases considerably(DSSim 42 % compared to Lily 15% and ASMOV 11 %). From the performance point of view the alignments took about 1.5 hour on a rather slow computer [5]. After the reviewing stage of the results we came to the conclusions that good results generation is challenging for all ontology pairs in this track.

3 General comments

3.1 Discussions on the way to improve the proposed system

We have experienced that developing ontology specific functionality into the mapping system could considerably improve the quality of the mappings. For example using aos:hasTranslation tag in the fao track can provide additional information for assessing similarities. However these solutions will only work for the specific ontologies only which contradicts with our objective to provide a good mapping system independent on the domain. From the background knowledge point of view we have concluded that based on the latest results that the additional multi lingual and domains specific background knowledge could provide added value for improving both recall and precision of the system.

[5] Pentium III 750 MHz, 512 MB

3.2 Comments on the OAEI 2008 procedure

The OAEI procedure and the provided alignment api works very well out of the box for the benchmarks, anatomy, directory, mldirectory and conference tracks. However for the fao, vlcr and library track we had to develop an SKOS parser, which can be integrated into the alignment api. Our SKOS parser convert SKOS file to OWL, which is then processed using the alignment api. Additionally we have developed a multi threaded chunk SKOS parser which can process SKOS file iteratively in chunks avoiding memory problems. For the vlcr track we had to develop several conversion and merging utility as the original file formats were not easily processable.

3.3 Comments on the OAEI 2008 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. The anatomy, library and mldirectory tests are perfect to verify the additional domain specific or multi lingual domain knowledge. Unfortunately this year we could not integrate our system with such background knowledge so the results are not as good as we expected.

4 Conclusion

Based on the experiments gained during OAEI 2006, 2007 and 2008 we had a possibility to realise a measurable evolution in our ontology mapping algorithm and test it with 8 different mapping tracks. Our main objective is to improve the mapping precision with managing the inherent uncertainty of any mapping process and information in the different ontologies. The different formalisms of the ontologies suggest that on the Semantic Web there is a need to qualitatively compare and evaluate the different mapping algorithms. 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. Further DSSim team was invited for oral presentation to the Ontology Mapping Workshop 2008 (OM-2008). An extract from the organizers is as follows: “Based on the discussion among the OAEI organisers and taking into account the number of tracks addressed and quality of matching results, it has been resolved that only the DSSim and ASMOV teams are offered to make oral presentations concerning their evalutation results”.

References

1. Shvaiko, P., Euzenat, J.: Ten challenges for ontology matching. Technical Report DISI-08-042, University of Trento (2008)
2. Nagy, M., Vargas-Vera, M., Motta, E.: Dssim - managing uncertainty on the semantic web. In: Proceedings of the 2nd International Workshop on Ontology Matching. (2007)

3. Shafer, G.: A Mathematical Theory of Evidence. (1976)
4. Nagy, M., Vargas-Vera, M., Motta, E.: Multi agent ontology mapping framework in the aqua question answering system. In: International Mexican Conference on Artificial Intelligence (MICAI-2005). (2005)
5. Besana, P.: A framework for combining ontology and schema matchers with dempster-shafer (poster). In: Proceedings of the International Workshop on Ontology Matching. (2006)
6. Yaghlane, B.B., Laamari, N.: Owl-cm: Owl combining matcher based on belief functions theory. In: Proceedings of the 2nd International Workshop on Ontology Matching. (2007)
7. Nagy, M., Vargas-Vera, M., Motta, E.: Feasible uncertain reasoning for multi agent ontology mapping. In: IADIS International Conference-Informatics 2008. (2008)
8. Nagy, M., Vargas-Vera, M., Motta, E.: Managing conflicting beliefs with fuzzy trust on the semantic web. In: The 7th Mexican International Conference on Artificial Intelligence (MICAI 2008). (2008)
9. Turney, P.D.: Similarity of semantic relations. Computational Linguistics **32**(3) (2006) 379–416
10. Kim, S.N., Baldwin, T.: Interpreting semantic relations in noun compounds via verb semantics. In: Proceedings of the COLING/ACL on Main conference poster sessions. (2006) 491–498
11. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence. (2003) 805–810

Results of GeRoMeSuite for OAEI 2008

Christoph Quix, Sandra Geisler, David Kensche, Xiang Li

Informatik 5 (Information Systems)
RWTH Aachen University, Germany
`http://www.dbis.rwth-aachen.de`

Abstract. *GeRoMeSuite* is a generic model management system which provides several functions for managing complex data models, such as schema integration, definition and execution of schema mappings, model transformation, and matching. The system uses the generic metamodel *GeRoMe* for representing models, and because of this, it is able to deal with models in various modeling languages such as XML Schema, OWL, ER, and relational schemas.

A component for schema matching and ontology alignment is also part of the system. We participated this year the first time in the OAEI contest in order to evaluate and compare the performance of our matcher component with other systems. Therefore, we focused our efforts on the 'benchmark' track.

1 Presentation of the system

Manipulation of models and mappings is a common task in the design and development of information systems. Research in Model Management aims at supporting these tasks by providing a set of operators to manipulate models and mappings. As a framework, *GeRoMeSuite* [4] provides an environment to simplify the implementation of model management operators. *GeRoMeSuite* is based on the generic role based metamodel *GeRoMe* [3], which represents models from different modeling languages (such as XML Schema, OWL, SQL) in a generic way. Thereby, the management of models in a polymorphic fashion is enabled, i.e. the same operator implementations are used regardless of the original modeling language of the schemas. In addition to providing a framework for model management, GeRoMeSuite implements several fundamental operators such as Match [7], Merge [6], and Compose [5].

The matching component of *GeRoMeSuite* has been described in more detail in [7], where we present and discuss in particular the results for heterogeneous matching tasks (e.g. matching XML Schema and OWL ontologies).

1.1 State, purpose, general statement

As a generic model management tool, *GeRoMeSuite* provides several matchers which can be used for matching models in general, i.e. our tool is not restricted to a particular domain or modeling language. Therefore, the tool provides several well known matching strategies, such as string matchers, Similarity Flooding, children and parent matchers, matchers using WordNet, etc. In order to enable the flexible combination of these basic matching technologies, matching strategies combining several matchers can be configured in a graphical user interface.

Because of its generic approach, *GeRoMeSuite* is well suited for matching tasks across heterogeneous modeling languages, such as matching XML Schema with OWL. We discussed in [7] that the use of a generic metamodel, which represents the semantics of the models to be matched in detail, is more advantageous for such heterogeneous matching tasks than a simple graph representation.

Furthermore, *GeRoMeSuite* is a holistic model management and not limited to schema matching or ontology alignment. It supports also other model management tasks such as schema integration [6], model transformation [2], mapping execution and composition [5].

1.2 Specific techniques used

The basis of *GeRoMeSuite* is the representation of models (including ontologies) in the generic metamodel *GeRoMe*. Any kind of model is transformed first into the generic representation, then the model management operators can be applied to the generic representation. The main advantage of this approach is that operators have to be implemented only once for the generic representation. In contrast to other (matching) approaches which use a graph representation without detailed semantics, our approach is based on the semantically rich metamodel *GeRoMe* which is able to represent modeling features in detail.

For the OAEI campaign, we focused on improving our matchers for the special case of ontology alignment, e.g. we added some features which are useful for matching ontologies. For example, the generic representation of models allows the traversal of models in several different ways. During the tests with the OAEI tasks, we realized that, in contrast to other modeling languages, traversing the ontologies using another structure than class hierarchy is not beneficial. Therefore, we configured all our matchers that take the model structure into account just to work with the class hierarchy. Furthermore, we implemented so called ‘children’ and ‘parent’ matchers, which propagate the similarity of elements up and down in the class hierarchy.

In addition, we also implemented a matcher using WordNet to discover synonyms in the ontologies. However, as the benchmark track contains only one example which uses synonyms, we did not include this matcher in the final configuration for the OAEI campaign.

1.3 Adaptations made for the evaluation

As only one configuration can be used for all matching tasks, we worked on strategies for measuring the quality of an alignment without having a reference alignment. We compared several statistical measures (such as expected value, variance, etc.) of alignments with different qualities in order to identify a ‘good’ alignment. Furthermore, these values can be used to set thresholds automatically.

During the tests, we made the experience that the expected value of all similarities, the standard deviation, and the number of mappings per model element can be used to evaluate the quality of an alignment.

Fig. 1 indicates the strategy which we used for the matching tasks in the benchmark track. The aggregation and filter steps are not fixed, they use the statistical values of the input similarities to adapt the actual values of, for example, threshold and aggregation weights.

The role matcher is a special matcher which compares the roles of model elements in our generic role-based metamodel. In principle, this results in that only elements of the same type are matched, e.g. classes with classes only and properties with properties only.

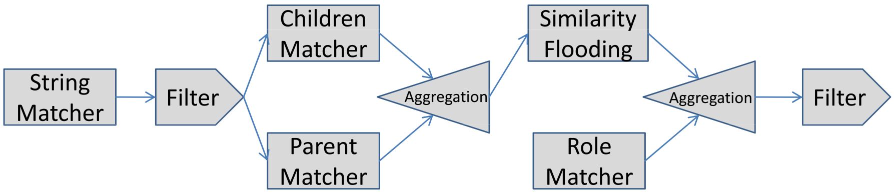


Fig. 1. Matching Strategy for OAEI

Furthermore, we experimented with histograms, i.e. a graphical representation of the distribution of similarity values. Although we could not identify particular patterns for histograms of 'good' or 'bad' alignments, we found the histogram quite useful for working interactively with the matching component of *GeRoMeSuite*. Fig. 2 shows a screenshot of *GeRoMeSuite*, including a window showing the histogram of a match result. In addition, the upper part of the window shows some statistical values for the current similarities. In another dialog, the filter can be adapted.

On a technical level, we implemented a command line interface for the matching component, as the matching component is normally used from within the GUI framework of *GeRoMeSuite*. The command line interface can work in a batch modus in which several matching tasks and configurations can be processed and compared.

1.4 Link to the system and parameters file

More information about the system can be found on the homepage of *GeRoMeSuite*: `http://www.dbis.rwth-aachen.de/gerome/`

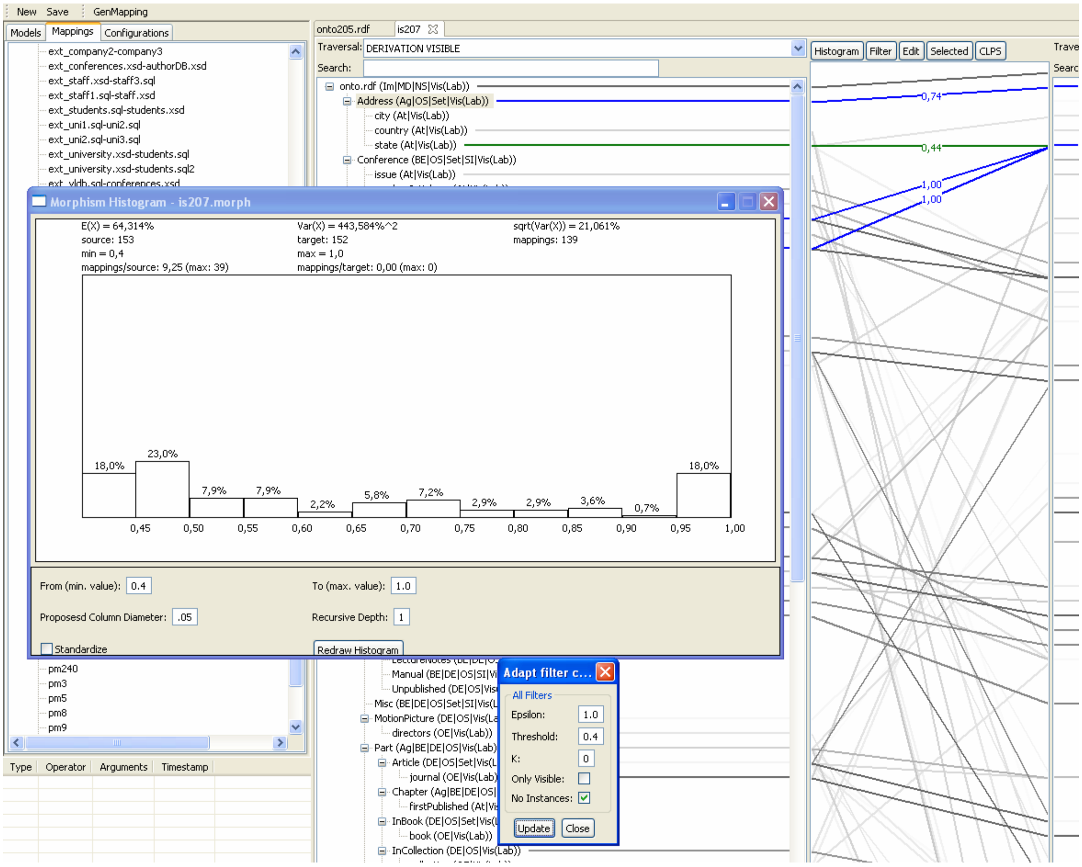
The page provides also links to the configuration files used for the evaluation.

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

The results for the OAEI campaign 2008 are available at `http://www.dbis.rwth-aachen.de/gerome/results.html`

2 Results

As we participated the first time in the OAEI campaign, we just focused on the benchmark track. The time used for each matching task was about 5 to 15 seconds.



Traversal: DERIVATION VISIBLE

Search:

E(X) = 64,314%   Var(X) = 443,584%^2   sqrt(Var(X)) = 21,061%
source: 153   target: 152   mappings: 139
min = 0,4   max = 1,0
mappings/source: 9,25 (max: 39)   mappings/target: 0,00 (max: 0)

From (min. value): 0.4   To (max. value): 1.0

Proposesd Column Diameter: .05   Recursive Depth: 1

□ Standardize

Epsilon: 1.0

Threshold: 0.4

K: 0

Only Visible: □

No Instances: ☑

Fig. 2. GUI of the Matching Component of *GeRoMeSuite* with Histogram Dialog

2.1 Benchmark

Overall, our matching component achieved very similar values for precision and recall, which seems to be rather unusual, if we compare our results with the results of other systems for previous years, where the precision was usually higher than recall.

Tasks 101-104 These tasks were quite easy as we could achieve a high precision and recall already with simple string matchers.

Task	Precision	Recall
101	1,00	0,79
103	0,94	0,79
104	0,95	0,79

For task 102 (irrelevant ontology), our matcher identified a few corresponding elements, such as year and yearValue, or date and year. Depending on the application of the mapping, such correspondences might be reasonable (e.g. for ontology merging).

Tasks 201-210 In these tasks, the linguistic information could not always be used as labels or comments were missing. After including also comments into the matching process, we could improve the match quality for these tasks significantly. For the synonym task (205), we also tested a matcher which uses WordNet to detect synonyms. However, as this matcher did not significantly improve the quality of the match result and required about three times more time than all other matchers together, we dropped the WordNet matcher from the final configuration.

Overall, the results are satisfying, except for the case 202, where no linguistic information at all was available.

Task	Precision	Recall
201	0,87	0,79
201-2	0,95	0,79
201-4	0,93	0,79
201-6	0,97	0,79
201-8	0,91	0,79
202	0,17	0,06
202-2	0,74	0,77
202-4	0,93	0,67
202-6	0,94	0,60
202-8	0,77	0,48
203	1,00	0,79
204	1,00	0,79
205	1,00	0,79
206	0,93	0,78
207	0,93	0,78
208	0,99	0,77
209	0,58	0,45
210	0,61	0,73

2.2 Tasks 221-231

The ontologies in these tasks lacked some structural information. As our matcher still uses string similarity in a first step, the results in this section were still quite reasonable.

Task	Precision	Recall
221	1,00	0,65
222	0,97	0,78
223	0,85	0,78
224	1,00	0,79
225	1,00	0,79
228	1,00	0,88
230	0,97	0,85
231	1,00	0,79

Tasks 232-266 These tasks are some combinations of the tasks before. For most of the tasks, the performance of our matcher was satisfying, but for some tasks, especially those without any linguistic information, it produced disappointing results. This gives some hints for future improvements of our matcher component, e.g. taking into account the overall structure of the ontology.

Tasks 301-304 For tasks 301 and 304, our system produce quite reasonable results. Further improvements could have been achieved, for example, by using the WordNet matcher for detecting synonyms, but we did not include this matcher because of performance reasons as explained above. Task 303 could not be processed by our system as there was a problem with importing this ontology into our generic representation.

3 Comments

A structured evaluation and comparison of ontology alignment and schema matching components is very useful for the development of such technologies. However, mappings between models are constructed for various reasons which can result in very different mapping results. For example, mappings for schema integration may differ from mappings for data translation. Therefore, different semantics for ontology alignments should be taken into account in the future, as it has been pointed out for schema matching in [1].

4 Conclusion

As our tool is neither specialized on ontologies nor limited to the matching task, we did not expect to deliver very good results. However, we were quite satisfied with the overall results. In general, we need to work on an improvement of the recall value. Furthermore, techniques used by other tools presented at the workshop would help us to improve the quality of the matching result and the performance of our tool. For example, identification of similar sub-structures in ontologies and semantic verification of the identified correspondences seem to be promising techniques to improve the quality and performance of the matching system.

Acknowledgements: This work is supported by the DFG Research Cluster on Ultra High-Speed Mobile Information and Communication (UMIC, `http://www.umic.rwth-aachen.de`).

References

1. J. Evermann. Theories of Meaning in Schema Matching: A Review. *Journal of Database Management*, **19**(3):55–82, 2008.
2. D. Kensche, C. Quix. Transformation of Models in(to) a Generic Metamodel. *Proc. BTW Workshop on Model and Metadata Management*, pp. 4–15. 2007.
3. D. Kensche, C. Quix, M. A. Chatti, M. Jarke. *GeRoMe*: A Generic Role Based Metamodel for Model Management. *Journal on Data Semantics*, **VIII**:82–117, 2007.
4. D. Kensche, C. Quix, X. Li, Y. Li. *GeRoMeSuite*: A System for Holistic Generic Model Management. C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C.-C. Kanne, W. Klas, E. J. Neuhold (eds.), *Proceedings 33rd Intl. Conf. on Very Large Data Bases (VLDB)*, pp. 1322–1325. Vienna, Austria, 2007.
5. D. Kensche, C. Quix, Y. Li, M. Jarke. Generic Schema Mappings. *Proc. 26th Intl. Conf. on Conceptual Modeling (ER'07)*, pp. 132–148. 2007.
6. C. Quix, D. Kensche, X. Li. Generic Schema Merging. J. Krogstie, A. Opdahl, G. Sindre (eds.), *Proc. 19th Intl. Conf. on Advanced Information Systems Engineering (CAiSE'07)*, LNCS, pp. 127–141. Springer-Verlag, 2007.
7. C. Quix, D. Kensche, X. Li. Matching of Ontologies with XML Schemas using a Generic Metamodel. *Proc. Intl. Conf. Ontologies, DataBases, and Applications of Semantics (ODBASE)*. 2007.

Lily: Ontology Alignment Results for OAEI 2008

Peng Wang, Baowen Xu

School of Computer Science and Engineering, Southeast University, China
{pwangseu@gmail.com}

Abstract. This paper presents the alignment results of Lily for the ontology alignment contest OAEI 2008. Lily is an ontology mapping system, and it has four main features: generic ontology matching, large scale ontology matching, semantic ontology matching and mapping debugging. In the past year, Lily has been improved greatly for both function and performance. In OAEI 2008, Lily submited the results for seven alignment tasks: benchmark, anatomy, fao, directory, mldirectory, library and conference. The specific techniques used by Lily are introduced briefly.The strengths and weaknesses of Lily are also discussed.

1 Presentation of the system

Currently more and more ontologies are distributedly used and built by different communities. Many of these ontologies would describe similar domains, but using different terminologies, and others will have overlapping domains. Such ontologies are referred to as heterogeneous ontologies, which is a major obstacle to realize semantic interoperation. Ontology mapping, which captures relations between ontologies, aims to provide a common layer from which heterogeneous ontologies could exchange information in semantically sound manners.

Lily is an ontology mapping system for solving the key issues related to heterogeneous ontologies, and it uses hybrid matching strategies to execute the ontology matching task. Lily can be used to discovery the mapping for both normal ontologies and large scale ontologies.

1.1 State, purpose, general statement

In order to obtain good alignments, the core principle of the matching strategy in Lily is utilizing the useful information effectively and rightly. Lily combines several novel and efficient matching techniques to find alignments. Currently, Lily realized four main functions: (1) Generic Ontology Matching method (GOM) is used for common matching tasks with small size ontologies. (2) Large scale Ontology Matching method (LOM) is used for the matching tasks with large size ontologies. (3) Semantic Ontology Matching method (SOM) is used for discovering the semantic relations between ontologies. Lily uses the web knowledge to recognize the semantic relations through the search engine. (4) Ontology mapping debugging is used to improve the alignment results.

The alignment process mainly contains three steps: (1) Preprocessing step parses the ontologies, and prepares the necessary data for the subsequent steps. (2) Match computing step uses suitable methods to compute the similarity between elements from different ontologies. (3)Post processing step is responsible for extracting, debugging and evaluating mappings. The architecture of Lily is shown in Fig. 1.

The lasted version of Lily is V2.0. Comparing with the last version V1.2, Lily has been enhanced greatly at both function and performance. Lily V2.0 provides a friendly graphical user interface. Fig.2 shows a snapshot when Lily is running.

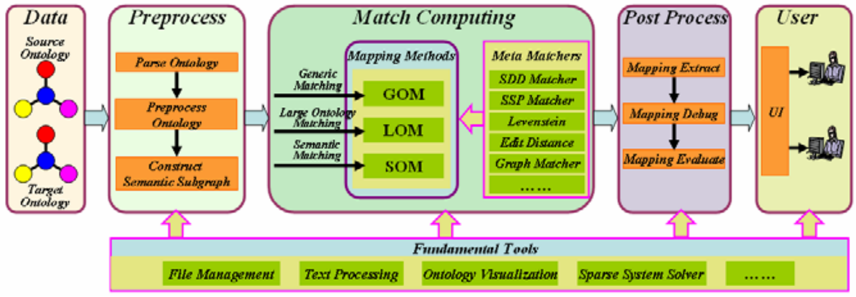
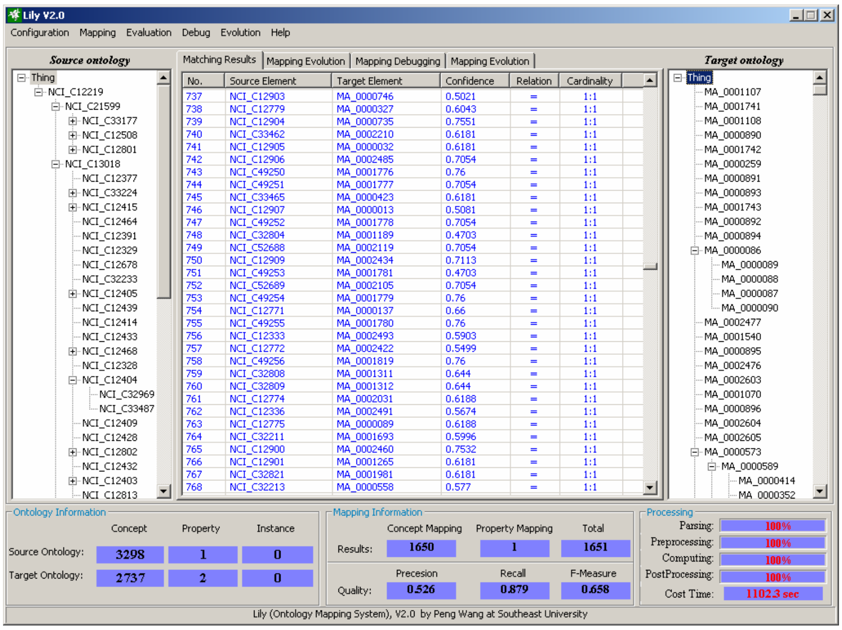


Fig. 1. The Architecture of Lily



Source ontology

- Thing
  - NCI_C12219
    - NCI_C21599
      - NCI_C33177
      - NCI_C12508
      - NCI_C12801
    - NCI_C13018
      - NCI_C12377
      - NCI_C33224
      - NCI_C12415
      - NCI_C12464
      - NCI_C12391
      - NCI_C12329
      - NCI_C12678
      - NCI_C32233
      - NCI_C12405
      - NCI_C12439
      - NCI_C12414
      - NCI_C12433
      - NCI_C12468
      - NCI_C12328
      - NCI_C12404
        - NCI_C32969
        - NCI_C33487
      - NCI_C12409
      - NCI_C12428
      - NCI_C12802
      - NCI_C12432
      - NCI_C12403
      - NCI_C12813

No.	Source Element	Target Element	Confidence	Relation	Cardinality
737	NCI_C12903	MA_0000746	0.5021	=	1:1
738	NCI_C12779	MA_0000327	0.6043	=	1:1
739	NCI_C12904	MA_0000735	0.7551	=	1:1
740	NCI_C33462	MA_0002210	0.6181	=	1:1
741	NCI_C12905	MA_0000032	0.6181	=	1:1
742	NCI_C12906	MA_0002485	0.7054	=	1:1
743	NCI_C49250	MA_0001776	0.76	=	1:1
744	NCI_C49251	MA_0001777	0.7054	=	1:1
745	NCI_C33465	MA_0000423	0.6181	=	1:1
746	NCI_C12907	MA_0000013	0.5081	=	1:1
747	NCI_C49252	MA_0001778	0.7054	=	1:1
748	NCI_C32804	MA_0001189	0.4703	=	1:1
749	NCI_C52688	MA_0002119	0.7054	=	1:1
750	NCI_C12909	MA_0002434	0.7113	=	1:1
751	NCI_C49253	MA_0001781	0.4703	=	1:1
752	NCI_C52689	MA_0002105	0.7054	=	1:1
753	NCI_C49254	MA_0001779	0.76	=	1:1
754	NCI_C12771	MA_0000137	0.66	=	1:1
755	NCI_C49255	MA_0001780	0.76	=	1:1
756	NCI_C12333	MA_0002493	0.5903	=	1:1
757	NCI_C12772	MA_0002422	0.5499	=	1:1
758	NCI_C49256	MA_0001819	0.76	=	1:1
759	NCI_C32808	MA_0001311	0.644	=	1:1
760	NCI_C32809	MA_0001312	0.644	=	1:1
761	NCI_C12774	MA_0002031	0.6188	=	1:1
762	NCI_C12336	MA_0002491	0.5674	=	1:1
763	NCI_C12775	MA_0000089	0.6188	=	1:1
764	NCI_C32211	MA_0001693	0.5996	=	1:1
765	NCI_C12900	MA_0002460	0.7532	=	1:1
766	NCI_C12901	MA_0001265	0.6181	=	1:1
767	NCI_C32821	MA_0001981	0.6181	=	1:1
768	NCI_C32213	MA_0000558	0.577	=	1:1

Target ontology

- Thing
  - MA_0001107
  - MA_0001741
  - MA_0001108
  - MA_0000890
  - MA_0001742
  - MA_0000259
  - MA_0000891
  - MA_0000893
  - MA_0001743
  - MA_0000892
  - MA_0000894
  - MA_0000086
    - MA_0000089
    - MA_0000088
    - MA_0000087
    - MA_0000090
  - MA_0002477
  - MA_0001540
  - MA_0000895
  - MA_0002476
  - MA_0002603
  - MA_0001070
  - MA_0000896
  - MA_0002604
  - MA_0002605
  - MA_0000573
    - MA_0000589
      - MA_0000414
      - MA_0000352

Ontology Information

	Concept	Property	Instance
Source Ontology:	3298	1	0
Target Ontology:	2737	2	0

Mapping Information

	Concept Mapping	Property Mapping	Total
Results:	1650	1	1651
	Precesion	Recall	F-Measure
Quality:	0.526	0.879	0.658

Processing

Parsing:	100%
Preprocessing:	100%
Computing:	100%
PostProcessing:	100%
Cost Time:	1102.3 sec

Lily (Ontology Mapping System), V2.0 by Peng Wang at Southeast University

Fig. 2. The user interface of Lily

1.2 Specific techniques used

Lily aims to provide high quality 1:1 alignments between concept/property pairs. The main specific techniques used by Lily are as follows.

Semantic subgraph An entity in a given ontology has its specific meaning. In our ontology mapping view, capturing such meaning is very important to obtain good alignment results. Therefore, before similarity computation, Lily first describes the meaning for each entity accurately. The solution is inspired by the method proposed by Faloutsos et al. for discovering connection subgraphs [1]. It is based on electricity analogues to extract a small subgraph that best captures the connections between two nodes of the graph. Ramakrishnan et al. also exploits such idea to find the informative connection subgraphs in RDF graph [2].

The problem of extracting semantic subgraphs has a few differences from Faloutsos's connection subgraphs. We modified and improved the methods provided by the above two work, and proposed a method for building an *n-size* semantic subgraph for a concept or a property in ontology. The subgraphs can give the precise descriptions of the meanings of the entities, and we call such subgraphs semantic subgraphs. The detail of the semantic subgraph extraction process is reported in our other work [3].

The significance of semantic subgraphs is that we can build more credible matching clues based on them. Therefore it can reduce the negative affection of the matching uncertain.

Generic ontology matching method The similarity computation is based on the semantic subgraphs, i.e. all the information used in the similarity computation is come from the semantic subgraphs. Lily combines the text matching and structure matching techniques [3].

Semantic Description Document (SDD) matcher measures the literal similarity between ontologies. A semantic description document of a concept contains the information about class hierarchies, related properties and instances. A semantic description document of a property contains the information about hierarchies, domains, ranges, restrictions and related instances. For the descriptions from different entities, we calculate the similarities of the corresponding parts. Finally, all separate similarities are combined with the experiential weights. For the regular ontologies, the SDD matcher can find satisfactory alignments in most cases.

To solve the matching problem without rich literal information, a similarity propagation matcher with strong propagation condition (SSP matcher) is presented, and the matching algorithm utilizes the results of literal matching to produce more alignments. Compared with other similarity propagation methods such as similarity flood [4] and SimRank [5], the advantages of our similarity propagation include defining stronger propagation condition, semantic subgraphs-based and with efficient and feasible propagation strategies. Using similarity propagation, Lily can find more alignments that cannot be found in the text matching process.

However, the similarity propagation is not always perfect. When more alignments are discovered, more incorrect alignments would also be introduced by the similarity propagation. So Lily also uses a strategy to determine when to use the similarity propagation.

Large scale ontology matching Large scale ontology matching tasks propose the rough time complexity and space complexity for ontology mapping systems. To solve this problem, we proposed a novel method [6], which uses the negative anchors and positive anchors to predict the pairs can be passed in the later matching computing. The method is different from other several large scale ontology matching methods, which are all based on ontology segment or modularization.

Semantic ontology matching Our semantic matching method [7] is base on the idea that Web is a large knowledge base, and from which we can gain the semantic relations between ontologies through Web search engine. Based on lexico-syntactic patterns, this method first obtains a candidate mapping set using search engine. Then the candidate set is refined and corrected with some rules. Finally, ontology mappings are chosen from the candidate mapping set automatically.

Ontology mapping debugging Lily uses a technique called ontology mapping debugging to improve the alignment results [8]. During debugging, some types of mapping errors, such as redundant and inconsistent mappings, can be detected. Some warnings, including imprecise mappings or abnormal mappings, are also locked by analyzing the features of mapping result. More importantly, some errors and warnings can be repaired automatically or can be presented to users with revising suggestions.

1.3 Adaptations made for the evaluation

In OAEI 2008, Lily used GOM matcher to compute the alignments for three tracks (benchmark, directory, conference). In order to assure the matching process is fully automated, all parameters are configured automatically with a strategy. For the large ontology alignment tracks (anatomy, fao, mldirectory, library), Lily used LOM matcher to discover the alignments. All parameters used by these tracks are same. Lily can determine which matcher should be chose according to the size of ontology.

1.4 Link to the system and the set of provided alignments

Lily V2.0 and the alignment results for OAEI 2008 are available at http://ontomappinglab.googlepages.com/lily.htm.

2 Results

2.1 benchmark

The benchmark test set can be divided into five groups: 101-104, 201-210, 221-247, 248-266 and 301-304.

101-104 Lily plays well for these test cases. But for the irrelevant ontology 102, Lily returns several alignments because it cannot decide whether the two ontologies are irrelevant, so it tries to find any possible alignments.

201-210 Lily can produce good results for this test set. Even without right labels and comments information, Lily can find most correct alignments through making use of other information such as instances. Using few alignment results obtained by the basic methods as inputs, the similarity propagation strategy will generate more alignments.

221-247 Lily can find most correct alignments using the labels and comments information.

248-266 This group is the most difficult test set. Lily first uses the SDD matcher to look for a few alignments. Then, using initial alignments as input, Lily exploits the SSP matcher to discover more alignments. In our experiments, too smaller and too bigger size semantic subgraph can not produce good alignments. *10-35* is a suitable size range in our experience. In 262, since almost all literal and structure information are suppressed, the similarity propagation can not find any results.

301-304 This test set are the real ontologies. Lily only finds the equivalent alignment relations.

The following table shows the average performance of each group and the overall performance on the benchmark test set.

Table 1. The performance on the benchmark

	101-104	201-210	222-247	248-266	301-304	Average	H-mean
Precision	1.00	1.00	0.99	0.93	0.86	0.95	0.97
Recall	1.00	0.95	1.00	0.76	0.79	0.84	0.88

2.2 anatomy

The anatomy track consists of two real large-scale biological ontologies. Lily can handle such ontologies smoothly with LOM method. Lily submitted the results for three sub-tasks in anatomy. Task#1 means that the matching system has to be applied with standard settings to obtain a result that is as good as possible. Task#2 means that the system generates the results with high precision. Task#3 means that the system generates the alignment with high recall.

Table 2 shows the performance of the task #1, #2 and #3 on anatomy test set, where Recall+ measures how many non trivial correct correspondences can be found in an alignment.

Table 2. The performance on the anatomy

	Runtime	Precision	Recall	Recall+	F-measure
Task#1	3h 20min	0.796	0.693	0.470	0.741
Task#2	3h 20min	0.863	0.640		0.664
Task#3	3h 20min	0.490	0.790	0.613	0.605

2.3 directory

The directory track requires matching two taxonomies describing the web directories. Except the class hierarchy, there is no other information in the ontologies. Therefore, besides the literal information, Lily also utilizes the hierarchy information to decide the alignments. Table 3 shows the performance on the directory test set.

Table 3. The performance on the directory

Precision	Recall	F-measure
0.59	0.37	0.46

2.4 conference

This task contains 15 real-case ontologies about conference. For a given ontology, we compute the alignments with itself, as well as with other ontologies. For we treat the equivalent alignment is symmetric, we get 105 alignment files totally. The heterogeneous character in this track is various. It is a challenge to generate good results for all ontology pairs in this test set.

The performance of Lily on this data set is shown as Table 4. The evaluation is based on two reference alignments.

Table 4. The performance on the conference based on reference mappings

	Precision	Recall	F-measure
Reference Alignment A	0.568	0.581	0.575
Reference Alignment B	0.432	0.500	0.463

2.5 fao

The task consists of several large scale ontologies about food and agricultural domain. The LOM method is used to find the alignments. Lily only provides the alignments between concepts or properties. Therefore, we did not submit the alignments for the subtask for finding the alignments between instances. Table 5 is the performance on fao data set.

Table 5. The performance on the fao

subtrack	Precision	Recall
agrafsa	0.867	0.403

2.6 library

This is a thesaurus mapping task. Lily only discovers the *extractMatch* alignments. Lily did not utilize the instance information provided in this year. Table 6 shows the evaluation results of Lily on this data set.

Table 6. The performance on the library

Evaluation Scenario	Precision	Coverage			
Thesaurus merging	0.529	0.368			
Annotation translation	Precision (book level)	Recall (book level)	Precision (annotation level)	Recall (annotation level)	Jaccard (annotation level)
	0.435	0.156	0.397	0.107	0.100

2.7 mldirectory

This task requires matching two web directories in different languages. For the reason that the ontologies provided by this task are hard to be parsed correctly, Lily only submits two alignment results for two subtasks (Auto and Movie) in English. Lily finds 377 alignments for Auto and 1864 alignments for Movie.

3 General comments

3.1 Comments on the results

Strengths For normal size ontologies, if they have regular literals or similar structures, Lily can achieve satisfactory alignments.

Weaknesses Lily needs to extract semantic subgraphs for all concepts and properties. It is a time-consuming process. Even though we have improved the efficiency of the extracting algorithm, it still is the bottleneck for the performance of the system.

4 Conclusion

We briefly introduce our ontology matching tool Lily. The matching process and the special techniques used by Lily are presented. The preliminary alignment results are carefully analyzed. Finally, we summarized the strengths and the weaknesses of Lily.

References

1. Faloutsos, C., McCurley, K. S., Tomkins, A.: Fast Discovery of Connection Subgraphs. In the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington (2004).
2. Ramakrishnan, C., Milnor, W. H., Perry, M., Sheth, A. P.: Discovering Informative Connection Subgraphs in Multirelational Graphs. ACM SIGKDD Explorations, Vol. 7(2), (2005)56-63.
3. Wang, P., Xu, B. A Generic Ontology Matching Method Based on Semantic Subgraph, to appear.
4. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In the 18th International Conference on Data Engineering (ICDE), San Jose CA (2002).
5. Jeh, G., Widom, J.: SimRank: A Measure of Structural-Context Similarity. In the 8th International Conference on Knowledge Discovery and Data Mining (SIGKDD), Edmonton, Canada, (2002).
6. Wang, P., Xu, B. A New Method for Matching Large Scale Ontologies, to appear.
7. Li, K., Xu, B., and Wang, P. An Ontology Mapping Approach Using Web Search Engine. Journal of Southeast University, 2007, 23(3):352-356.
8. Wang, P., Xu, B. Debugging Ontology Mapping: A Static Method. Computing and Informatics, 2008, 27(1): 21–36.

Appendix: Raw results

The final results of benchmark task are as follows.

Matrix of results

#	Comment	Prec.	Rec.	#	Comment	Prec.	Rec.
101	Reference alignment	1.00	1.00	251		0.96	0.76
103	Language generalization	1.00	1.00	251-2		0.99	0.96
104	Language restriction	1.00	1.00	251-4		0.99	0.90
201	No names	1.00	1.00	251-6		0.94	0.83
201-2		1.00	1.00	251-8		0.99	0.85
201-4		1.00	1.00	252		0.96	0.79
201-6		1.00	1.00	252-2		0.97	0.94
201-8		1.00	1.00	252-4		0.97	0.94
202	No names, no comment	1.00	0.84	252-6		0.97	0.94
202-2		1.00	0.97	252-8		0.97	0.94
202-4		1.00	0.92	253		0.81	0.59
202-6		0.98	0.87	253-2		0.98	0.93
202-8		0.98	0.85	253-4		1.00	0.92
203	Misspelling	1.00	1.00	253-6		0.95	0.81
204	Naming conventions	1.00	1.00	253-8		0.95	0.79
205	Synonyms	1.00	0.99	254		1.00	0.27
206	Translation	1.00	0.99	254-2		1.00	0.82
207		1.00	0.99	254-4		1.00	0.70
208		1.00	0.99	254-6		1.00	0.61
209		0.97	0.88	254-8		1.00	0.42
210		1.00	0.89	257		0.50	0.06
221	No hierarchy	1.00	1.00	257-2		1.00	0.97
222	Flattened hierarchy	1.00	1.00	257-4		0.94	0.88
223	Expanded hierarchy	0.98	0.98	257-6		0.84	0.79
224	No instances	1.00	1.00	257-8		0.89	0.76
225	No restrictions	1.00	1.00	258		0.80	0.60
228	No properties	1.00	1.00	258-2		0.97	0.94
230	Flattening entities	0.94	1.00	258-4		0.96	0.88
231	Multiplying entities	1.00	1.00	258-6		0.95	0.82
232	No hierarchy no instance	1.00	1.00	258-8		0.94	0.78
233	No hierarchy no property	1.00	1.00	259		0.89	0.70
236	No instance no property	1.00	1.00	259-2		0.98	0.95
237		1.00	1.00	259-4		0.98	0.95
238		0.99	0.99	259-6		0.98	0.95
239		0.97	1.00	259-8		0.98	0.95
240		0.97	1.00	260		0.94	0.55
241		1.00	1.00	260-2		0.96	0.93
246		0.97	1.00	260-4		0.93	0.93
247		0.94	0.97	260-6		0.96	0.79

248		1.00	0.81	260-8		0.88	0.72
248-2		1.00	0.95	261		0.67	0.48
248-4		1.00	0.92	261-2		0.88	0.91
248-6		1.00	0.88	261-4		0.88	0.91
248-8		0.98	0.85	261-6		0.88	0.91
249		0.83	0.66	261-8		0.88	0.91
249-2		0.98	0.95	262		NaN	0.00
249-4		0.98	0.91	262-2		1.00	0.79
249-6		0.98	0.87	262-4		1.00	0.61
249-8		0.95	0.82	262-6		1.00	0.42
250		0.90	0.58	262-8		1.00	0.21
250-2		1.00	1.00	265		0.80	0.14
250-4		1.00	1.00	266		0.30	0.09
250-6		1.00	1.00	301	BibTeX/MIT	0.94	0.82
250-8		1.00	0.88	302	BibTeX/UMBC	0.89	0.65
				303	Karlsruhe	0.65	0.71
				304	INRIA	0.95	0.97

MapPSO Results for OAEI 2008

Jürgen Bock[1] and Jan Hettenhausen[2]

[1] FZI Research Center for Information Technology, Karlsruhe, Germany
bock@fzi.de
[2] Griffith University, Institute for Integrated and Intelligent Systems, Brisbane, Australia
j.hettenhausen@griffith.edu.au

Abstract. We present first results of an ontology alignment approach that is based on discrete particle swarm optimisation. In this paper we will firstly describe, how the algorithm approaches the ontology matching task as an optimisation problem, and briefly sketch how the specific technique of particle swarm optimisation is applied. Secondly, we will briefly discuss the results gained for the Benchmark data set of the 2008 Ontology Alignment Evaluation Initiative.

1 Presentation of the system

We introduce the Ontology **Map**ping by **P**article **S**warm **O**ptimisation (MapPSO) system as a novel research prototype, which is expected to become a highly scalable, massively parallel tool for ontology alignment. In the following subsection the basic idea of this approach will be sketched.

1.1 State, purpose, general statement

The MapPSO algorithm is being developed for the purpose of aligning large ontologies. Instance mapping however is not part of our efforts. Motivated by the observation that ontologies and schema information such as thesauri or dictionaries are not only getting numerous on the web, but also are becoming increasingly large in terms of the number of classes/concepts and properties/relations. This development raises the need for highly scalable tools to provide interoperability and integration of various heterogeneous sources. On the other hand the emergence of parallel architectures provide the basis for highly parallel and thus scalable algorithms which need to be adapted to these architectures.

For the presented MapPSO method we formulated the ontology alignment problem as an optimisation problem which allowed us to employ a discrete variant of particle swarm optimisation [1, 2], a population based optimisation paradigm inspired by social interaction between swarming animals. Particularly the population based structure of this method provides high scalability on parallel systems. Particle swarm optimisation furthermore belongs to the group of anytime algorithms, which allow for interruption at any time and will provide the best answer being available at that time. Particularly this property might be interesting when an alignment problem is subject to certain time constraints.

1.2 Specific techniques used

MapPSO utilises a discrete particle swarm optimisation (DPSO) algorithm, based in parts on the DPSO developed by Correa *et al.* [1, 2], to tackle the ontology matching problem as an optimisation problem. The core element of this optimisation problem is the objective function which supplies a fitness value for each candidate alignment.

To find solutions for the optimisation problem, MapPSO simulates a set of particles whereby each particle is a candidate alignment comprising a set of initially random mappings[3]. Each of these particles maintains a memory of previously found good mappings (*personal best*) and the swarm maintains a collective memory of the best known alignment so far (*global best*). In each iteration, particles are updated by changing their sets of correspondences in a guided random manner. Correspondences which are also present in the *global best* set are more likely to be kept, as are those with a very good evaluation. In addition the number of correspondences represented by each particle also changes according to the number of correspondences in the *global best* alignment in a self-adaptation process.

Each candidate alignment of two ontologies is scored based on a weighted sum of quality measures of the single correspondences, and the number of correspondences it consists of. The currently best alignment is the one with the best known fitness rating according to these criteria. According to this revisit of the ontology matching problem, a particle swarm can be applied to search for the optimal alignment.

For each correspondence the quality score is calculated based on an aggregation of scores from a configurable set of base matchers. Each base matcher provides a distance measure for each correspondence. Currently the following well known base matchers are used:

- SMOA string distance [3] for entity names
- SMOA string distance for entity labels
- WordNet distance for entity names
- WordNet distance for entity labels
- Vector space similarity [4] for entity comments
- Hierarchy distance to propagate similarity of superclasses / superproperties
- Structural similarity of classes derived from properties that have them as domain or range classes
- Structural similarity of properties derived from their domain and range classes

For each correspondence the available base distances are aggregated by applying the OWA operator [5]. The OWA operator performs an **O**rdered **W**eighted **A**verage aggregation of the base distances by ordering the base distances and applying a fixed weight vector. The evaluation of the overall alignment of each particle is computed by aggregating all its correspondence distances and accounting for the number of correspondence represented by this particle.

In the current implementation each of the particles runs in an individual thread and all fitness calculations and particle updates are performed in parallel. The only sequential portion on the algorithm is the synchronisation after each iteration to acquire the fitness value from each particle and determine the currently global best alignment.

[3] Currently only 1:1 alignments are supported.

1.3 Adaptations made for the evaluation

Since MapPSO is an early prototype, we did use the OAEI 2008 Benchmark test data during the development process. No specific adaptations have been made.

1.4 Link to the system and parameters file

The release of MapPSO for OAEI 2008 is located in the package `MapPSO` at
`http://ontoware.org/projects/mappso/`

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

The alignment results of MapPSO for the Benchmark test case of OAEI 2008 are located in the package `alignResults` at
`http://ontoware.org/projects/mappso/`

2 Results

Since MapPSO is in an early development stage, we only participate in the Benchmark test case in the OAEI 2008.

2.1 benchmark

The Benchmark test case is designed to provide a number of data sets systematically revealing strengths and weaknesses of the matching algorithm. In the case of MapPSO the experiences were as follows:

The MapPSO algorithm is highly adjustable via its parameter file and can be tuned to perform well on specific problems, as well as to perform well for precision or recall. To obtain the results presented in table 1 we used a compromised parameter configuration.

For **tests 101-104** MapPSO achieves precision values of around 90 % and recall values of 100 %. Test 102 with a totally irrelevant ontology, however, still determines a number of wrong correspondences.

As for **tests 201-210** results are not as positive, as the quality of the alignment decreases with the number of features that provide linguistic features to exploit. For test case 202 where all names and comments are unavailable, MapPSO performs worst in this group of tests.

In **tests 221-247**, where the structure of the ontologies varies, the results are similar to the 10x tests. Since the main focus of the current implementation of MapPSO's base matchers is on linguistic features, such as string distance and WordNet distance.

The **tests 248-266** combine linguistic and structural problems. As the results show, the quality of the alignments is decreasing with the decreasing number of features available in the ontologies.

For the real-life cases, **tests 301-304**, no uniform results can be derived as the algorithm's precision and recall values vary between 0 and 60 %.

Table 1. MapPSO results for benchmark test cases.

Test Name	Precision	Recall	Test Name	Precision	Recall	Test Name	Precision	Recall
101	0.9	1	241	0.79	1	254-8	0.71	0.15
102	0	NaN	246	0.81	1	257	0.05	0.06
103	0.94	1	247	0.73	0.82	257-2	0.91	0.61
104	0.92	1	248	0.04	0.04	257-4	0.53	0.61
201	0.12	0.13	248-2	0.75	0.79	257-6	0.4	0.52
201-2	0.79	0.88	248-4	0.48	0.54	257-8	0.23	0.27
201-4	0.66	0.7	248-6	0.36	0.4	258	0.08	0.09
201-6	0.5	0.56	248-8	0.16	0.18	258-2	0.74	0.74
201-8	0.28	0.31	249	0.06	0.07	258-4	0.49	0.53
202	0.05	0.05	249-2	0.73	0.82	258-6	0.34	0.39
202-2	0.72	0.81	249-4	0.53	0.59	258-8	0.2	0.23
202-4	0.55	0.6	249-6	0.34	0.38	259	0.01	0.01
202-6	0.34	0.37	249-8	0.16	0.18	259-2	0.68	0.76
202-8	0.2	0.23	250	0.07	0.09	259-4	0.64	0.72
203	0.95	0.94	250-2	0.78	0.85	259-6	0.66	0.74
204	0.85	0.93	250-4	0.67	0.48	259-8	0.66	0.73
205	0.3	0.33	250-6	0.38	0.48	260	0.03	0.03
206	0.35	0.38	250-8	0.21	0.27	260-2	0.67	0.76
207	0.35	0.39	251	0.07	0.08	260-4	0.53	0.72
208	0.78	0.88	251-2	0.76	0.8	260-6	0.64	0.31
209	0.22	0.25	251-4	0.47	0.53	260-8	0.21	0.28
210	0.18	0.2	251-6	0.28	0.3	261	0.04	0.06
221	0.9	1	251-8	0.22	0.24	261-2	0.86	0.36
222	0.91	1	252	0.06	0.06	261-4	0.82	0.27
223	0.96	0.89	252-2	0.62	0.7	261-6	0.75	0.45
224	0.9	1	252-4	0.63	0.71	261-8	0.68	0.79
225	0.9	1	252-6	0.63	0.69	262	0.07	0.09
228	0.8	1	252-8	0.63	0.71	262-2	0.86	0.76
230	0.86	1	253	0.06	0.07	262-4	0.5	0.55
231	0.92	1	253-2	0.75	0.71	262-6	0.79	0.33
232	0.94	1	253-4	0.5	0.56	262-8	0.16	0.21
233	0.79	1	253-6	0.38	0.42	265	0.03	0.03
236	0.8	1	253-8	0.17	0.19	266	0.02	0.03
237	0.93	1	254	0	0	301	NaN	0
238	0.9	0.95	254-2	0.85	0.7	302	0.22	0.21
239	0.89	0.86	254-4	0.83	0.45	303	NaN	0
240	0.71	0.82	254-6	0.37	0.39	304	0.65	0.64

3 General comments

In the following we will provide a few statements on our experiences from participating in the OAEI 2008 competition and briefly discuss future work on the MapPSO algorithm.

3.1 Comments on the results

Firstly it shall be noted that MapPSO is a non-deterministic method and therefore on a set of independent runs the quality of the results and the number of mappings in the alignments will be subject to slight fluctuations.

For many of the benchmark test cases the current implementation of MapPSO could already provide reasonably good solutions. However, particularly alignments which are largely based on structural criteria currently impose a problem on the algorithm and require further development such as the addition of appropriate base matchers. This behaviour is particularly reflected in test cases, where lexical and linguistic information is omitted, such as in 201 and 202.

The submitted results were furthermore all acquired with an identical configuration file with a non-optimised and rather general set of parameters. For individual alignment problems, the quality of fitness values and thereby to some extend the efficiency of the algorithm can be improved by limiting the selection of base matchers to those that are most likely to provide useful ratings for the involved ontologies.

3.2 Discussions on the way to improve the proposed system

One of the most crucial component of MapPSO is the acquisition of fitness values for individual mappings and complete alignments. The MapPSO algorithm currently uses various base matchers, which are, in the current release naively implemented. It can be assumed that improving the current base matchers as well as adding further base matchers for an extended set of criteria will be highly beneficial for MapPSO. This regards in particular the aforementioned problem of taking structural properties of the alignments into account.

In addition, various other optimisations and extensions to the algorithm are conceivable. Particularly the extension of self-adaptation to the weight parameters and further optimisation of the currently implemented self-adapting length of candidate alignments appear to be promising. We hope to participate in next year's OAEI campaign demonstrating better performance on the benchmark test case and providing results for additional larger test cases on which we can demonstrate the scalability of the MapPSO approach.

4 Conclusion

In this paper we briefly introduced our ontology alignment system MapPSO and some results for the OAEI 2008 competition. Despite the fact that MapPSO is still at an early stage of development we could achieve promising results for the majority of the benchmark alignments. Key features of the discrete particle swarm optimisation approach of MapPSO are high parallel scalability and the possibility to either set time constraints for the alignment or interrupt the alignment process at any time and acquire the best alignment MapPSO could find up to that point. Future work on MapPSO will focus on improving the weighting and scoring methods of the fitness function and improve usage of structural information of the ontologies as a mean of calculating score values for candidate alignments.

Acknowledgement

The presented research was partially funded by the German Federal Ministry of Economics (BMWi) under the project Theseus (number 01MQ07019). We would furthermore like to extend our gratitude to Florian Berghoff for his contributions to the implementation of the method.

References

1. Correa, E.S., Freitas, A.A., Johnson, C.G.: A New Discrete Particle Swarm Algorithm Applied to Attribute Selection in a Bioinformatics Data Set. In: Proceedings of the 8th Genetic and Evolutionary Computation Conference (GECCO-2006), New York, NY, USA, ACM (2006) 35–42
2. Correa, E.S., Freitas, A.A., Johnson, C.G.: Particle Swarm and Bayesian Networks Applied to Attribute Selection for Protein Functional Classification. In: Proceedings of the 9th Genetic and Evolutionary Computation Conference (GECCO-2007), New York, NY, USA, ACM (2007) 2651–2658
3. Stoilos, G., Stamou, G., Kollias, S.: A String Metric For Ontology Alignment. In: Proceedings of the 4rd International Semantic Web Conference. Volume 3729 of LNCS., Galway, Ireland, Springer (November 2005) 624–637
4. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM **18**(11) (1975) 613–620
5. Ji, Q., Haase, P., Qi, G.: Combination of Similarity Measures in Ontology Matching using the OWA Operator. In: Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Base Systems (IPMU'08). (2008)

RiMOM Results for OAEI 2008

Xiao Zhang[1], Qian Zhong[1], Juanzi Li[1], Jie Tang[1], Guotong Xie[2] and Hanyu Li[2]
[1]Department of Computer Science and Technology, Tsinghua University, China
{zhangxiao,zhongqian,ljz,tangjie}@keg.cs.tsinghua.edu.cn
[2]IBM China Research Lab
{XIEGUOT, lihanyu}@cn.ibm.com

Abstract. In this report, we give a brief explanation of how RiMOM obtains the ontology alignment results at OAEI 2008 contest. We introduce the alignment process of RiMOM and more than 8 different alignment strategies integrated in RiMOM. Since every strategy is defined based on one specific ontological-information, we, in particular, study how the different strategies perform for different alignment tasks in the contest and design a strategy selection technique to get better performance. The result shows this technique is very useful. We also discuss some future work about RiMOM.

1 Presentation of the system

Ontology matching is the key technology to reach interoperability over ontologies. In recent years, much research work has been conducted for finding the alignment of ontologies [1] [2].

RiMOM [3] is an automatic ontology matching system implemented in JAVA. In RiMOM, we implement several different matching strategies. Each strategy is defined based on one kind of ontological information. Moreover, we investigate the differences between the strategies and compare the performances of different strategies on different matching tasks. One of the most important issues we introduce in RiMOM is how to choose appropriate strategies (or strategy combination) according to the features and the information of the ontologies.

1.1 State, purpose, general statement

For simplifying the following description, we here define the notations used throughout the report. An ***ontology*** *O* is composed of concepts *C*, properties/relations *R*, instances *I*, and Axioms A^o. We here use capital letters to indicate a set and lowercase letters (e.g. $c \in C$) to indicate one element in the set. Sometimes, for further simplification, we use entity *e* to indicate either *c* or *r*.

We implement more than 8 different strategies in RiMOM. Experiments show that the multi-strategy based alignments do not always beat its single strategy counterpart. We define three ontology feature factors: Label Similarity Factor (LF), Structure Similarity Factor (SF) and Label Meaning Factor (MF) for strategy selection. The definition of the three factors can be found in 1.2.1.

There are six major steps in a general alignment process of RiMOM.

1) Ontology feature factors estimation. Given two ontologies, it estimates three ontology feature factors. The three factors are used in the next step of strategy selection.

2) Strategy selection. The basic idea of strategy selection is that 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 base strategies on them. Moreover, if the labels are full of semantic, we will use WordNet [4] based strategy instead of Edit-distance based strategies. We also use these factors to decide the thresholds when refining the results. Strategy selection is mainly used on the benchmark data set. For the directory, mldirectory, anatomy, and fao data set, we choose the strategies manually.

3) Single 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 a similarity propagation process to refine the found alignments and to find new alignments that cannot be found using other strategies.

6) Alignment refinement. It refines the alignment results from the previous steps. We defined several heuristic rules to remove the "unreliable" alignments.

1.2 Specific techniques used

1.2.1 Ontology Feature factors estimation

Given two ontologies: source Ontology O_1 and target ontology O_2, we calculate three ontology feature factors, including two approximate similarity factors between two ontologies (Structure Similarity Factor and Label Similarity Factor) and one factor representing the semantics of entity labels in each ontology (Label Meaning Factor) .

We define structure similarity factor as: $SF = \frac{\#common\_concept}{\max(\#nonleaf\_O_1, \#nonleaf\_O_2)}$, where $\#nonleaf\_O_1$ indicates the number of concepts in O_1 that has sub concepts. Likewise for $\#nonleaf\_O_2$. $\#common\_concept$ 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 root concept, we add one to $\#common\_concept$. After enumerating 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: $LS = \frac{\#same\_label}{\max(\#c_1, \#c_2)}$, 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 that have the same label.

The label meaning factor is defined as: $MF = \frac{\#label\_with\_meaning}{\#entity}$, where $\#label\_with\_meaning$ represents the number of entities whose label is meaningful, and $\#entity$ represents the number of entities in the ontology. We use WordNet to judge whether a label is meaningful or not.

Now the three factors are defined very simply. The first two factors are 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 Factor MF is larger than 0.9, then RiMOM uses WordNet based strategy instead of edit-distance based strategy. If the structure similarity factor SF is lower than 0.25, then RiMOM suppresses the CCP and PPP strategies. However, the CPP will always be used in the alignment process.

1.2.2 Multiple strategies

The strategies implemented in RiMOM include: edit-distance based strategy, vector-based similarity based strategy, path-similarity based strategy, dynamic path-similarity based strategy, Japanese-English path-similarity strategy and similarity-propagation based strategies.

1. Edit-distance based strategy(ED)

Each label (such as concept names or property names) is composed of several tokes. In this strategy, we calculate the edit distance between labels of two entities. Edit distance estimates the number of operation needed to convert one string into another. We define $(1 - \#op / \max\_length(l(e_1), l(e_2)))$ as the similarity of two labels, where $\#op$ indicates the number of operations, $\max\_length(l(e_1), l(e_2))$ represents the maximal length of the two labels.

2. WordNet based strategy (WN)

In this strategy, RiMOM first preprocesses each label into a bag of words. When calculate the similarity from one bag of words to another, for every word in the first bag, RiMOM find the most similar word in the second with WordNet, then calculate the mean of the similarities as the similarity from the first bag to the second. The similarity of the two labels is the mean of the similarity of two bags of words in two directions.

3. Vector-similarity based strategy(VS)

We formalize the problem as that of document similarity. For any entity *e*, we regard its label, comment, and instances as a "document" and calculate the similarity between entities. Specially, the "document" is tokenized into words. Then we remove the stop words and employ stemming on the words and view the remains as features to generate a feature vector. 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 the root concept. Next, we compute the cosine similarity between two feature vectors. The advantage of this strategy is that it can easily incorporate different information (even structural information) into the feature vector.

4. Path-similarity based strategy (PS)

We define the path of labels as the aggregation of the entity labels from the root entity to the current entity. The paths of the labels of the two entities can be represented as $L_1 = a_1 a_2 .. a_m$ and $L_2 = b_1 b_2 .. b_n$. The path-similarity measure between two entities *e₁* and *e₂* is defined as:

$$Sim(e_1, e_2) = \sum_{i=1}^{m-1} w_i \bullet \max_{j=1}^{n-1}(LabelSim(a_i, b_j)) + w_m \bullet LabelSim(a_m, b_n))$$

The $LabelSim(a_i, b_j)$ is calculated using either edit-distance or WordNet.

5. Dynamic path-similarity based strategy (DPS)

The path of labels can also be considered as a path of entities, especially when the main information of the ontology is the labels. We have three assumption for this strategy: 1) for the two path of entities, we always match from the short path to the long one, and every entity in the short path can be matched to an entity in the long one; 2) no matched pairs are "crossed", that is to say, the matching result is consistent with the hierarchy represented in the path; 3) when calculating the similarity of current pair of entities, the matching result of the prev-path is optimal. Then we can calculate the similarity of two paths of entities using the dynamic programming technique.

6. Strategy combination

For some alignment task, we need use more than one strategy to find the alignment. The strategies are employed first independently to calculate the similarity between entities and the similarities are combined together. Our combination measure is defined as:

$$Sim(e_1, e_2) = \frac{\sum_{k=1}^{n} w_k \sigma\left(Sim_k(e_1, e_2)\right)}{\sum_{k=1}^{n} w_k}$$

Where $e_1 \in O_1$ and $e_2 \in O_2$. $Sim_k(e_1, e_2)$ is the alignment score obtained by strategy *k*. w_k is the weight of strategy *k*. For vector similarity based strategy, the weight is always 1 while for WordNet and edit-distance based strategies, the weight is generated automatically. σ is sigmod function, which is defined as $\sigma(x) = 1/\left(1 + e^{-5(x-\alpha)}\right)$, where α is tentatively set as 0.5.

7. Similarity-propagation 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 [7], we define three structure based strategies in RiMOM, namely concept-to-concept propagation strategy (CCP), property-to-property propagation strategy (PPP), and the concept-to-property propagation strategy (CCP).

Intuition of the propagation based method is that if two entities are aligned, their super-concepts have higher probability to be aligned. The basic idea here is to propagate the similarity of two entities to entity pairs that have relations (e.g.

subClassOf, superClassOf, siblingClassOf, subPropertyOf, superPropertyOf, range and domain) with them. The idea is inspired by similarity flooding [8]. We extended the algorithm and adaptively used them in the three structure based strategies.

In CCP, we propagate similarities of concepts pair across the concept hierarchical structure. Likely, we propagate similarities of property pair across the property hierarchy in PPP and concepts pair to their corresponding property pair in CPP.

Furthermore, there are some object properties in the ontologies which may have the similar characteristics with subClassOf property. Every pair of concepts with such property has a relation similar to sub-super concept relation. However, these pairs of concepts are usually manipulated as the domain and range of property and the relation is lost. RiMOM can also use these properties for similarity-propagation.

The similarity-propagation based strategies are performed after other strategies defined above. They can be used to adjust the alignments and find new alignments.

8. Indirect Matching

We also use the indirect matching technique in RiMOM. It is sometimes very difficult to match two ontologies directly. Since the source ontology and the target ontology are usually concerned with the same domain of knowledge, it is possible to match both the source and target ontology to a third one. Then the entities in the source and target ontology which match to the same entity in the third ontology can be aligned. RiMOM can take three ontologies as input and execute the indirect matching.

1.3 Adaptations made for the evaluation

Some parameters are tuned and set in the experiments. For example, for strategy selection, we define 0.25 as threshold to determine whether CCP and PPP will be suppressed or not. We also define MF factor threshold as 0.9 to determine whether use WordNet based strategy instead of edit-distance based strategy. In addition, we employ dynamic path similarity for directory task and path-similarity based strategy for mldirectory task.

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

Our system RiMOM (RiMOM does not need the parameters file) can be found at http://keg.cs.tsinghua.edu.cn/project/RiMOM/.
The alignment results of the campaign are available at
http://keg.cs.tsinghua.edu.cn/project/RiMOM/OAEI2008/.

2 Results

RiMOM has participated in 5 tasks in OAEI 2008, including benchmark, anatomy, fao, directory and mldirectory. RiMOM use OWL-API to parse the RDF and OWL files. The experiments are carried out on a PC running Window XP with AMD Athlon 64 X2 4200+ processor (2.19GHz) and 2G memory.

2.1 benchmark

There are in total 111 alignment tasks defined on the benchmark data set. RiMOM takes exactly the same steps introduced in 1.1. However, on the tasks where the labels are absolutely random strings, the WordNet based strategy and edit-distance based strategy are suppressed. The vector-similarity based strategy is always employed.

RiMOM get perfect alignment in the 101, 103, 104 tests. RiMOM also do quite well in the 2xx tests. Except the data sets in which almost all the information are suppressed like 26x and 25x, RiMOM aligns the source and target ontology with both good precision and recall. Even in those data set most information missing, RiMOM still can find some alignments with very high precision. Compared to the result of OAEI 2007, RiMOM also improve the performance in the real ontology data sets 301, 302, 303, 304.

2.2 anatomy

The anatomy data set contains two large scale anatomy ontologies. RiMOM employs edit-distance based strategy on labels to get the initial mapping, then employs both the concept-to-concept propagation strategy and the propagation strategy on the object property "UNDEFINED_part_of" to get the alignments which cannot be extracted by just comparing the labels simply. The propagation strategy can find about 15% more alignments.

2.3 fao

The scale of the fao data set is even larger than the anatomy data set, so we only use the edit-distance based strategy on labels to calculate the similarity. Moreover, because the FAO ontology is better formed than larger than the other two, we use the FAO ontology as a standard ontology to indirectly match the AGROVOC ontology and ASFA ontology.

2.4 directory

As all the ontologies in directory data set are in the "chain" form, RiMOM just employs the dynamic path-similarity based strategy to get the similarity matrix. Then RiMOM extracts the alignments with no "crossed" matched entity pairs.

2.5 mldirectory

The mldirectory data set is composed of three kinds of tasks: the matching between English ontologies, the matching between Japanese ontologies and English ontologies and the matching between the Japanese ontologies. For this task, RiMOM mainly depends on the ID of the concepts and the hierarchical information. When dealing with the Japnanese IDs, we takes the following preprocessing steps: 1) use the tool named ChaSen [5] for segmentation of Japanese IDs; 2) use the dictionary JMDict [6] to translate the Japanese words into English; 3) for those Japanese words in katakana which cannot be found in JMDict, convert them into their Roman spelling. Through this we get the corresponding English IDs for these Japanese IDs. Then we use the path-similarity based strategy to align these ontologies.

3 General comments

3.1 Comments on the results

From the results we can see that RiMOM can take advantage of all kinds of information on the ontologies to achieve high performance. The linguistic information is especially important for RiMOM. The structure information and the instance information make a good improvement on the results. When the linguistic information is not available (for example, when the labels of entities are meaningless), the structure information and other information are very important.

Strategy selection is effective in the alignment process. With strategy selection, RiMOM can avoid some noise produced by some strategies when the information these strategies rely on is not adequate. This is a very interesting issue: how to find the best strategy (combination) for a specific matching task. Although we add the MF factor this year compared to last year, it is far from the ideal solution for the strategy selection problem.

We adjust some refinement strategies this year and this change is very helpful in the real ontology matching problem. We also re-implement some of our propagation strategies to make them more efficient so they can be applied on the large scale tasks. With these improvements, RiMOM performs better on large scale data sets such as anatomy and fao.

Since the cross-lingual matching tasks are introduced this year, we make a trial on the process of Japanese ontologies and get a fairly good result. We think the cross-lingual task is very important in ontology matching.

3.2 Discussions on the way to improve the proposed system

First of all, we are very eager to improve our strategy selection mechanism. There are two major issues: 1) what are the essential features of an ontology and what are the essential similarity features between ontologies? How should we describe these features? 2) How to do the strategy selection automatically and more effectively based on these features.

Secondly, we will improve the capability of RiMOM to deal with large scale ontologies. Up to now most strategies in RiMOM cannot be applied to large scale ontologies because of memory and time limit. The vlcr task of OAEI 2008 will be a great challenge.

3.3 Comments on the OAEI 2008 test cases

The benchmark test is better defined than OAEI 2007. The data set is very interesting and makes it easy to find the strength and weakness of matching systems. It is very helpful for us to improve our system.

The mldirectory data set is very interesting. It is a very good challenge to deal with the multi-lingual ontology matching tasks.

In the directory data set, however, there may be conflicts in the "chain" hierarchy. That is to say, there are concepts with more than 1 super-concepts and sub-concepts. We think the problem comes from that a folder may have a sub folder with the same name. When extracting the ontologies, the folder and its same-named folder are given the same URI.

4 Conclusion

In this report, we have briefly introduced how we employed RiMOM to obtain the alignment results in OAEI'08 contest. We have presented the alignment process of RiMOM and explained the strategy defined in RiMOM. We have also described how we performed the alignment for different alignment tasks. We summarized the strengths and the weaknesses of our proposed approach and make our comments on the results.

Acknowledgments. The work is supported by the National Natural Science Foundation of China (90604025), Chinese National Key Foundation Research and Development Plan (2007CB310803), and IBM SUR project.

References

1. Euzenat, J., Shivaiko. P.: Ontology Matching. Springer-Verlag, Berlin-Heidelberg, 2007.
2. Kalfoglou, Y., Schorlemmer, M.: Ontology Maching: The State of the Art. The Knowledge Engineering Review Journal, 2003.
3. Tang, J., Li, J., Liang, B., Huang, X., Li, Y., and Wang, K.: Using Bayesian Decision for Ontology Alignment. Journal of Web Semantics, Vol(4) 4, pp. 243-262, 2006.
4. http://wordnet.princeton.edu/
5. http://chasen-legacy.sourceforge.jp/
6. http://www.csse.monash.edu.au/~jwb/j_jmdict.html
7. Felzenszwalb, P.F. and Huttenlocher, D.P.: Efficient belief propagation for early vision. International Journal of Computer Vision, Vol. 70, No. 1, October 2006
8. Melnik, S., Garcia-Molina, H. and Rahm, E.: 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

SAMBO and SAMBOdtf Results for the Ontology Alignment Evaluation Initiative 2008

Patrick Lambrix, He Tan, and Qiang Liu

Department of Computer and Information Science
Linköpings universitet
581 83 Linköping, Sweden

Abstract. This article describes a base system for ontology alignment, SAMBO, and an extension, SAMBOdtf. We present their results for the benchmark, anatomy and FAO tasks in the 2008 Ontology Alignment Evaluation Initiative. For the benchmark and FAO tasks SAMBO uses a strategy based on string matching as well as the use of a thesaurus. It obtains good results in many cases. For the anatomy task SAMBO uses a combination of string matching and the use of domain knowledge. This combination performed well in former evaluations using other anatomy ontologies. SAMBOdtf uses the same strategies but, in addition, uses an advanced filtering technique that augments recall while maintaining a high precision.

1 Presentation of the system

In this section we present the purpose of SAMBO and SAMBOdtf, the framework on which they are built, the specific techniques that are used and the adaptations made for the evaluation.[1]

1.1 State, purpose, general statement

Although several of our methods and techniques are general and applicable to different areas, when developing SAMBO, we have focused on biomedical ontologies. We chose this field because ontologies are recognized as important in some of the grand challenges in the biomedical domain, and many biomedical ontologies have been developed and are publicly available and have overlapping information. This has, however, had an influence on the approaches on which we focused. In general, ontologies may contain concepts, relations, instances and axioms. Most biomedical ontologies are controlled vocabularies, taxonomies, or thesauri. This means that they may contain concepts, is-a and part-of relations, and sometimes a limited number of other relationships. Therefore, we have mainly developed methods that are based on these ontology components. For some approaches we have also used documents about a concept as instances for that concept. We have not dealt with axioms. SAMBOdtf is an extension of SAMBO that uses an advanced filtering method.

[1] Some parts of the description of the system are the same as last year's description in [11].

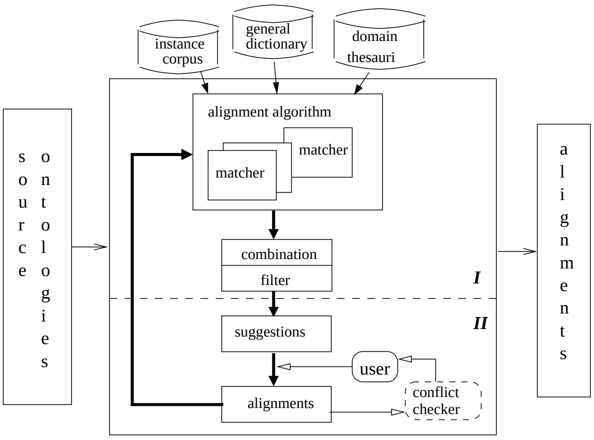


Fig. 1. Alignment framework [4].

1.2 Framework

SAMBO and SAMBOdtf are based on the framework shown in figure 1 [4]. The framework consists of two parts. The first part (*I* in figure 1) computes alignment suggestions. The second part (*II*) interacts with the user to decide on the final alignments. An alignment algorithm receives as input two source ontologies. The algorithm includes one or several matchers, which calculate similarity values between the terms from the different source ontologies. The matchers may use knowledge from different sources. Alignment suggestions are then determined by combining and filtering the results generated by one or more matchers. By using different matchers and combining and filtering the results in different ways we obtain different alignment strategies. The suggestions are then presented to the user who accepts or rejects them. The acceptance and rejection of a suggestion may influence further suggestions. Further, a conflict checker is used to avoid conflicts introduced by the alignment relationships. The output of the alignment algorithm is a set of alignment relationships between terms from the source ontologies.

1.3 Specific techniques used

In this section we describe the matchers, and combination and filtering techniques that are available in SAMBO and SAMBOdtf. These matchers and techniques were previously evaluated using test cases for aligning Gene Ontology and Signal Ontology, and for aligning Medical Subject Headings (MeSH) and the Anatomical Dictionary for the Adult Mouse (MA) [4] using the KitAMO evaluation environment [5].[2] In addition to these techniques we have also experimented with other matchers [7, 9, 12]. We are also working on methods for recommendation of alignment strategies [10] which we intend to integrate into SAMBO in the future.

[2] An introduction to SAMBO and KitAMO can be found in [6].

Matchers SAMBO and SAMBOdtf contain currently five basic matchers: two terminological matchers, a structure-based matcher, a matcher based on domain knowledge, and a learning matcher. We describe the matchers used in OAEI-2008, and mention the others briefly.

Terminological matchers. The basic terminological matcher, *Term* contains matching algorithms based on the textual descriptions (names and synonyms) of concepts and relations. In the current implementation, the matcher includes two approximate string matching algorithms, n-gram and edit distance, and a linguistic algorithm. An n-gram is a set of n consecutive characters extracted from a string. Similar strings will have a high proportion of n-grams in common. Edit distance is defined as the number of deletions, insertions, or substitutions required to transform one string into the other. The greater the edit distance, the more different the strings are. The linguistic algorithm computes the similarity of the terms by comparing the lists of words of which the terms are composed. Similar terms have a high proportion of words in common in the lists. A Porter stemming algorithm is employed to each word. These algorithms were evaluated in [3] using MeSH anatomy (ca 1400 terms) and MA (ca 2350 terms). Term computes similarity values by combining the results from these three algorithms using a weighted sum. The combination we use in our experiments (weights 0.37, 0.37 and 0.26 for the linguistic algorithm, edit distance and n-gram, respectively) outperformed the individual algorithms in our former evaluations [3]. Further, the matcher *TermWN* is based on Term, but uses a general thesaurus, WordNet (http://wordnet.princeton.edu/), to enhance the similarity measure by looking up the hypernym relationships of the pairs of words in WordNet.

Structural matcher. The structural matcher is an iterative algorithm based on the is-a and part-of hierarchies of the ontologies. The algorithm requires as input a list of alignment relationships and similarity values and can therefore not be used in isolation. The intuition behind the algorithm is that if two concepts lie in similar positions with respect to is-a or part-of hierarchies relative to already aligned concepts in the two ontologies, then they are likely to be similar as well.

Use of domain knowledge. Another strategy is to use domain knowledge. Our matcher *UMLSKSearch* uses the Metathesaurus in the Unified Medical Language System (UMLS, http://www.nlm.nih.gov/research/umls/). The similarity of two terms in the source ontologies is determined by their relationship in UMLS. In our experiments we used the UMLS Knowledge Source Server to query the UMLS Metathesaurus with source ontology terms. The querying is based on searching the normalized string index and normalized word index provided by the UMLS Knowledge Source Server. We used version 2008AA of UMLS. As a result we obtain concepts that have the source ontology term as their synonym. We assign a similarity value of 1[3] if the source ontology terms are synonyms of the same concept and 0 otherwise.[4]

[3] For the anatomy task we assign a value of 0.99 in order to introduce a preference of exact string matching over UMLSKSearch. Although this is not useful for SAMBO, it is used in the adaptations made especially for OAEI.

[4] Observe that this is slightly different from the version reported in [4] where we used version 2005AA of UMLS and we assigned a similarity value of 1 for two terms with the exact same names, 0.6 if the source ontology terms are synonyms of the same concept, and 0 otherwise.

Learning matcher. The matcher makes use of life science literature that is related to the concepts in the ontologies. It is based on the intuition that a similarity measure between concepts in different ontologies can be defined based on the probability that documents about one concept are also about the other concept and vice versa.

Combinations The user is given the choice to employ one or several matchers during the alignment process. The similarity values for pairs of concepts are then determined based on the similarity values computed by one matcher, or as a weighted sum of the similarity values computed by different matchers.

Filtering The filtering method in SAMBO is single threshold filtering. Pairs of concepts with a similarity value higher than or equal to a given threshold value are returned as alignment suggestions to the user.

SAMBOdtf implements the double threshold filtering method developed in [1]. The double threshold filtering approach uses the structure of the ontologies. It is based on the observation that (for the different approaches in the evaluation in [4]) for single threshold filtering the precision of the results is decreasing and the recall is increasing when the thresholds are decreasing. Therefore, we propose to use two thresholds. Pairs with similarity value equal or higher than the upper threshold are retained as suggestions. The intuition is that this gives suggestions with a high precision. Further, pairs with similarity values between the lower and the upper threshold are filtered using structural information and the rest is discarded. We require that the pairs with similarity values between the two thresholds are 'reasonable' from a structural point of view.[5] The intuition here is that the recall is augmented by adding new suggestions, while at the same time the precision stays high because only structurally reasonable suggestions are added. The double threshold filtering approach contains the following three steps. (i) Find a consistent suggestion group from the pairs with similarity value higher or equal than the upper threshold. We say that a set of suggestions is a consistent suggestion group if each concept occurs at most once as first argument in a pair, at most once as second argument in a pair and for each pair of suggestions (A,A') and (B,B') where A and B are concepts in the first ontology and A' and B' are concepts in the second ontology: A $\subset$ B iff A' $\subset$ B'. (ii) Use the consistent suggestion group to partition the original ontologies. (iii) Filter the pairs with similarity values between the lower and upper thresholds using the partitions. Only pairs of which the elements belong to corresponding pieces in the partitions are retained as suggestions. For details we refer to [1].

1.4 Adaptations made for the evaluation

SAMBO and SAMBOdtf are interactive alignment systems. The alignment suggestions calculated by SAMBO and SAMBOdtf are normally presented to the user who accepts or rejects them. Alignment suggestions with the same concept as first item in the pair are shown together to the user. Therefore, the systems show the user the different alternatives for aligning a concept. This is a useful feature, in particular when the system

[5] In our implementation we have focused on the is-a relation.

computes similarity values which are close to each other and there is no or only a small preference for one of the suggestions. Further, the acceptance and rejection of a suggestion may influence which suggestions are further shown to the user.

The computation of the alignment suggestions in SAMBO and SAMBOdtf is based on the computation of a similarity value between the concepts. The computation of the similarity values does not take into account what the relationship of the alignment should be. However, when an alignment is accepted, the user can choose whether the alignment relationship should be an equivalence relation or an is-a relation.

As the OAEI evaluation only considers the non-interactive part of the system and the computation of the similarity values does not take the relationship into account, we had to modify the computation of the suggestions. It would not make sense to have alignment suggestions where a concept appears more than once as the user would not be able to make a choice. Therefore, we decided to filter our systems' alignment suggestion lists such that only suggestions are retained where the similarity between the concepts in the alignment suggestion is higher than or equal to the similarity of these concepts to any other concept according to the alignment suggestion list. (In the case there are different possibilities, one is randomly chosen. In the implementation the first in the list is chosen.)

1.5 Link to the system and parameters file

The SAMBO (and SAMBOdtf) project page is at
http://www.ida.liu.se/∼iislab/projects/SAMBO/.

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

The suggested alignments are available at
http://www.ida.liu.se/∼iislab/projects/SAMBO/OAEI/2008/.

2 Results

We have provided alignment suggestions for the tasks ’benchmark’, ’anatomy’ and ’FAO’. Tests were performed on a IBM R61i Laptop, WinXP Intel(R) Pentium(R) Dual T2370 @ 1.73GHz, 1.73GHz, 1.99G RAM.

2.1 Benchmark

For the benchmark task the results for SAMBO were obtained by using TermWN with threshold 0.6. We introduced a preprocessing step where we used two strategies to generate names and for each case used the one that gave the best result for TermWN. The first strategy splits names based on capital letters occurring within a name. For instance, ’InCollection’ was split into ’In Collection’. In the second strategy we remove stop words such that, for instance, ’is part of’ is converted into ’part’. We did not use the comment field. The results may be improved using also this field.

We assume that ontology builders use a reasonable naming scheme and thus we did not tackle the cases where labels were replaced by a random one. Therefore, the recall for tests 201-202, 248-254, 257-262, 265-266. For these cases we may use other kind of information in the ontology such as the comment field or the structure. For the tests that were new for this year [*-2,4,6,8] where the labels are scrambled, the precision is high. In general, the recall is high when few of the labels are scrambled and drops when more labels are scrambled. We also did not focus on different natural languages (206-207, 210) or subsumption relationships (302).

Regarding the other cases we received high precision and recall except for cases 205 and 209. For 205 and 209 we had expected that using WordNet would be an advantage. Therefore, we compared the results with a run using Term (without WordNet). The differences between the results for Term and TermWN were small for all cases, including cases 205 and 209.

For SAMBOdtf we used the same matcher with upper threshold 0.8 and lower threshold 0.4. In the cases where there is no is-a hierarchy, SAMBOdtf with upper threshold 0.8 gives the same results as SAMBO with threshold 0.8. This is also the case when there are no suggestions with similarity value above 0.8, or no suggestions with similarity value between 0.4 and 0.8. Most of the test cases for benchmark belonged to one of these categories. For other test sets, we got the same result as SAMBO for [252-2,4,6,8], [259-2,4,6,8], [261-2,4,6,8], and 301. We obtained a little better recall for 204-210 and 304, since the lower threshold introduced some new alignments, most of which were correct.

2.2 Anatomy

Task 1 The results for the anatomy task for SAMBO were obtained by first running exact string matching and retaining the pairs with similarity values 1. On the remainder we run UMLSKSearch and retain the pairs with similarity value at least 0.99. Finally, we run TermWN[6] with threshold 0.6 on the remainder of the pairs. With respect to the computation of the suggestions, this would be similar to having a matcher that returns as similarity value for a pair the maximum of the similarity value for the pair according to UMLSKSearch and the similarity value for the pair according to TermWN, and then using 0.6 as threshold. SAMBO generated 1465 alignment suggestions. SAMBO reached a precision of 0.869, a recall of 0.836 and an f-value of 0.852. Further, it reached a recall+ of 0.586. This was the best result for all 9 participating systems in OAEI 2008.[7] In 2007 we used a version of SAMBO that used Term instead of TermWN and a previous version of UMLS. The 2007 version reached a better recall for non-trivial alignments, but at the cost of an overall decrease of precision and recall. A possible explanation for this is our strategy for choosing maximum one alignment suggestion per concept. In 2008 exact matching strings were preferred, while in 2007 there was no preference between pairs that had exact matching strings or pairs that were proposed based on domain knowledge.

[6] Last year we used Term instead of TermWN.

[7] The system with best f-measure in 2007 obtained 0.928 precision, 0.815 recall, 0.523 recall+ and 0.868 f-measure.

For SAMBOdtf, the same strategy is used, but with upper threshold 0.8 and lower threshold 0.4. SAMBOdtf generates 1527 alignment suggestions. Of these suggestions, 1440 have a similarity value between 0.6 and 0.8. This means that SAMBOdtf filtered out 25 of the suggestions obtained by SAMBO with threshold 0.6. (A manual check seems to suggest that most of these are correctly filtered out, but some are wrongly filtered out.) Further, SAMBOdtf also filtered out 19 suggestions with similarity values between 0.4 and 0.6. (A manual check seems to suggest that these were correctly filtered out.) SAMBO reached a precision of 0.831, a recall of 0.833, an f-value of 832 and a recall+ of 0.579. This was the second best result for all 9 participating systems in OAEI 2008.

The running time for SAMBO was ca 12 hours and for SAMBOdtf ca 17 hours.

Task 4 For task 4, we augmented SAMBO and SAMBOdtf in the following ways.

For SAMBO we added the alignments in the partial reference list to the list of alignment suggestions, but with a special status. These alignments could not be removed in the special filtering step that was introduced for OAEI (see section 1.4). SAMBO generated 1494 suggestions of which 988 are also in the partial reference list. SAMBO obtained the best results of the participating systems. With respect to the unknown part of the reference alignment, its precision in increased with 0.024, its recall decreased with 0.002 and its f-value increased with 0.011

For SAMBOdtf we also added the alignments in the partial reference list to the list of alignment suggestions with the special status. In addition, we used the partial reference list in the double threshold filtering step. We used a consistent part[8] of the partial reference list as a consistent suggestion group. For upper threshold 0.8 and lower threshold 0.4 we obtained 1547 alignment suggestions. SAMBOdtf obtained the second best results of the participating systems. With respect to the unknown part of the reference alignment, its precision increased with 0.040, its recall with 0.008 and its f-value with 0.025. SAMBOdtf was the system with the highest increase in f-value and was the only system that used the partial reference alignment to increase both precision and recall. This result is most likely due to the fact that, in contrast to task 1 where the consistent suggestion group consists of suggestions, in this task the consistent suggestion group consists of true alignments. Therefore, the suggestions with similarity value between the two thresholds that are retained are structurally reasonable with respect to true alignments and not just (although with high confidence) suggestions.

2.3 FAO

We only show results for the first task in FAO. For SAMBO we used TermWN with threshold 0.6. For SAMBOdtf we used TermWN with upper threshold 0.8 and lower threshold 0.4.

3 General comments

A problem that users face is that often it is not clear how to get the best alignment results given that there are many strategies to choose from. In most systems (including

[8] The partial reference list is actually not a consistent group.

ours) there usually is no strategy for choosing the matchers, combinations and filters in an optimal way. Therefore, we used our experience from previous evaluations [4, 1] to decide which matchers and thresholds to use for which task. The lack of an optimization strategy is also the reason why we did not provide results for the second and third test for anatomy (optimization with respect to precision and recall, respectively). In the future, however, this may be possible using recommendation methods for alignment strategies such as proposed in [10] that will be able to recommend matchers, combinations and filters based on the alignment task and evaluation methods.

The OAEI deals with the non-interactive part of the alignment systems. This allows for evaluating how good the alignment suggestions are. However, for some systems, such as SAMBO and SAMBOdtf, the list of alignment suggestions is only an initial list and is updated after each acceptance or rejection of a suggestion.

4 Conclusion

We have briefly described our ontology alignment systems SAMBO and SAMBOdtf and some results of them on the alignment tasks of OAEI.

For the benchmark task we have used TermWN and obtained good results in many cases. We expect that the results will still improve when we use more information available in the ontology, such as the comment field and the structure.

Regarding the anatomy task we have used a combination of UMLSKSearch and TermWN, which performed best in former evaluations using other anatomy ontologies. We are currently also evaluating instance-based matchers [7].

A major problem is deciding which algorithms should be used for a given alignment task. This is a problem that users face, and that we have also faced in the evaluation. We expect that recommendation strategies [10, 8, 2] will alleviate this problem.

References

1. B Chen, H Tan, and P Lambrix. Structure-based filtering for ontology alignment. In *Proceedings of the IEEE WETICE Workshop on Semantic Technologies in Collaborative Applications*, pages 364–369, 2006.
2. M Ehrig, S Staab, and Y Sure. Bootstrapping ontology alignment methods with APFEL. In *Proceedings of the International Semantic Web Conference*, pages 186–200, 2005.
3. P Lambrix and H Tan. Merging DAML+OIL ontologies. In J Barzdins and A Caplinskas, editors, *Databases and Information Systems - Selected Papers from the Sixth International Baltic Conference on Databases and Information Systems*, pages 249–258. IOS Press, 2005.
4. P Lambrix and H Tan. SAMBO - a system for aligning and merging biomedical ontologies. *Journal of Web Semantics, Special issue on Semantic Web for the Life Sciences*, 4(3):196–206, 2006.
5. P Lambrix and H Tan. A tool for evaluating ontology alignment strategies. *Journal on Data Semantics, LNCS 4380*, VIII:182–202, 2007.
6. P Lambrix and H Tan. Ontology alignment and merging. In Burger, Davidson, and Baldock, editors, *Anatomy Ontologies for Bioinformatics: Principles and Practice*, pages 133–150. Springer, 2008.

7. P Lambrix, H Tan, and W Xu. Literature-based alignment of ontologies. In *Proceedings of the Third International Workshop on Ontology Matching*, 2008.
8. M Mochol, A Jentzsch, and J Euzenat. Applying an analytic method for matching approach selection. In *Proceedings of the Workshop on Ontology Matching*, 2006.
9. H Tan, V Jakonienė, P Lambrix, J Aberg, and N Shahmehri. Alignment of biomedical ontologies using life science literature. In *Proceedings of the International Workshop on Knowledge Discovery in Life Science Literature, LNBI 3886*, pages 1–17, 2006.
10. H Tan and P Lambrix. A method for recommending ontology alignment strategies. In *Proceedings of the 6th International Semantic Web Conference, LNCS 4825*, pages 494–507, 2007.
11. H Tan and P Lambrix. SAMBO results for the ontology alignment evaluation initiative 2007. In *Proceedings of the Second International Workshop on Ontology Matching*, pages 236–243, 2007.
12. T Wächter, H Tan, A Wobst, P Lambrix, and M Schroeder. A corpus-driven approach for design, evolution and alignment of ontologies. In *Proceedings of the Winter Simulation Conference*, pages 1595–1602, 2006. Invited contribution.

Spider:
Bringing Non-Equivalence Mappings to OAEI

Marta Sabou[1] and Jorge Gracia[2]

[1] Knowledge Media Institute (KMi), The Open University, UK
r.m.sabou@open.ac.uk
[2] IIS Department, University of Zaragoza, Spain
jogracia@unizar.es

Abstract. With the large majority of existing matching systems focusing on deriving equivalence mappings, OAEI has been primarily focused on assessing such kind of relations. As the field inevitably advances towards the discovery of more complex mappings, the contest will need to reflect such changes as well. In this paper we present Spider, a system that provides alignments containing not only equivalence mappings, but also a variety of different mapping types (namely, subsumption, disjointness and named relations). Our goal is both to get an insight into the functioning of our system and, more importantly, to assess the current support for dealing with non-equivalence mappings in the OAEI contest. We hope that our observations will contribute to further enhance the procedure of the contest.

1 Presentation of the system

1.1 State, purpose, general statement

Our purpose was to investigate two concrete issues related to non-equivalence mappings.

1. Do non-equivalence mappings bring a good addition to alignments made up of purely equivalence mappings? We have investigated this question during OAEI'07 without being able to draw a clear conclusion. During that contest we submitted an alignment made up of only non-equivalence mappings to the FAO food task. While the expert evaluation gave a general insight in the performance of the matcher itself, due to the large size of the dataset it was impossible to draw a conclusion on whether this alignment was a useful increment to simply equivalence mappings. Also, such a study was hampered by the fact that our system contained only non-equivalence mappings. Based on these lessons, this year we have teamed up with a matcher which provides equivalent mappings only. Also, and most importantly, we have restricted our study to the smallest test of the contest, the benchmark test set. This should give us a clearer understanding on the amount and quality of non-equivalence mappings that can be discovered in addition to equivalence mappings. Such results, if positive, will motivate us (and hopefully others) in building hybrid systems which can go beyond equivalence mappings.

2. Is the OAEI procedure capable of handling non-equivalence mappings? Since the majority of matching systems focus on equivalence mappings, the OAEI contest is currently geared towards evaluating such mappings. However, as the field inevitably evolves towards more complex mappings, this will probably have an impact on OAEI as well. We want to assess the current support for evaluating non-equivalence mappings and, based on our experience, offer our ideas about potential future improvements.

1.2 Specific techniques used

Our system combines two concrete subsystems. First, the CIDER algorithm is used to derive equivalence mappings. Second, this alignment is extended with non-equivalence mappings derived by Scarlet.

CIDER, also described in this workshop, uses a semantic similarity measure to compare the concepts of the two input ontologies. This schema-based method combines different elementary techniques, as linguistic similarities or vector space modelling, to compare the ontological context of each of the involved terms. The discovered correspondences that score below a certain threshold are filtered out of the resultant alignment. This measure has been adapted from the authors' earlier work on word sense disambiguation [6]. More details about CIDER are provided in [2].

Scarlet [5] automatically selects and explores online ontologies *to discover relations between two given concepts*. For example, when relating two concepts labeled *Researcher* and *AcademicStaff*, Scarlet 1) identifies (at run-time) online ontologies that can provide information about how these two concepts inter-relate and then 2) combines this information to infer their relation. In [5] the authors describe two increasingly sophisticated strategies to identify and to exploit online ontologies for relation discovery. Hereby, we rely on the first strategy that derives a relation between two concepts if this relation is defined within a single online ontology, e.g., stating that *Researcher* $\sqsubseteq$ *AcademicStaff*. Besides subsumption relations, Scarlet is also able to identify disjoint and named relations. All relations are obtained by using derivation rules which explore not only direct relations but also relations deduced by applying subsumption reasoning within a given ontology. For example, when matching two concepts labeled *Drinking Water* and *tap_water*, appropriate anchor terms are discovered in the TAP ontology and the following subsumption chain in the external ontology is used to deduce a subsumption relation: *DrinkingWater* $\sqsubseteq$ *FlatDrinkingWater* $\sqsubseteq$ *TapWater*.

1.3 Link to the system and parameters file

The version of CIDER used for this evaluation can be found at `http://sid.cps.unizar.es/SEMANTICWEB/ALIGNMENT/OAEI08/`

Scarlet can be accessed online and downloaded from: `http://scarlet.open.ac.uk/`

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

Our results can be found at `http://kmi.open.ac.uk/people/marta/oaei/SPIDER.zip`.

2 Results

We have focused on test sets 3xx as these propose the comparison of real-life ontologies and contain a few non-equivalence mappings in the reference alignment. The rest of the tests in this set do not make sense for Scarlet as comparison is sought between modified versions of the same ontology.

2.1 Results Computed by Organizers

The evaluation of the benchmark alignments consists in an automatic comparison to a manually built reference alignment. The reference alignments for cases 3xx contain mostly equivalence mappings. The alignments for cases 301, 302 and 303 also contain a few subsumption relations between the matched ontologies but these are not enough to evaluate a significant part of our alignment which contains non-equivalences. A good way to practically demonstrate this is to compare the results obtained by CIDER and Spider. As it is visible from Table 1, despite the fact that the second alignment is more complex, numerically speaking, the results are worse. Indeed, as expected, while recall increases for those cases where the reference alignments also contain subsumption relations, precision is heavily affected.

Test Set	CIDER		Spider	
	Prec.	**Rec.**	**Prec.**	**Rec.**
301	0.88	0.59	0.27	0.67
302	0.94	0.60	0.26	0.75
303	0.81	0.79	0.08	0.81
304	0.95	0.95	0.16	0.95

Table 1. Results computed for the 3xx tests by comparison to the reference alignments.

We think that the current evaluation should be improved to better accommodate non-equivalence mappings, because, as we will see in the next sections, such mappings can bring an important addition to alignments made up only of equivalences.

2.2 Results Obtained with Other Modalities

We have performed a manual evaluation of the non-equivalence mappings obtained for the 3xx benchmark tests. Given the simplicity of the domain, the evaluation was performed by a single person, one of the authors. Therefore we regard these results as indicative only until a more extended multi-evaluator evaluation will be performed.

Test Set	**Total mappings**	**True mappings**	**True redundant**	**True - non redundant**	**False**	**Overall Precision**	**Core Precision**
301	112	71	30	41	41	63%	50%
302	116	64	11	53	52	55%	50%
303	458	233	84	149	225	51%	40%
304	386	255	128	127	131	66%	50%

Table 2. Results for the manual evaluation of the 3xx benchmark tests.

For each alignment we have assessed the true and false mappings. In the case of true mappings, we differentiate between redundant and non-redundant mappings. Redundant mappings are those mappings which could be deduced by considering the source ontologies and the equivalence alignment. Formally: $mapping_r \models (O_s, O_t, A_=)$. Obviously, one can argue that these mappings are of little interest as they could be easily deduced.

Consequently, we compute two kinds of precision values. First, the overall precision takes into account all true mappings, whether redundant or not. Second, the core precision excludes the redundant mappings and considers only the non-redundant ones.

The results are shown in Table 2. The overall precision of the alignment is in the range of 50% and 70% thus correlating to earlier findings performed in different domains [5]. If we do not take redundant mappings into account, the precision of the remaining alignment drops to an average of 50%. This shows that, on average, at least half of the mappings in the extended alignment are correct and thus bring an addition to the purely equivalence based mappings. The number of non-redundant true mappings shows the net increment that this tool brings to the equivalence based alignment. Even for small ontologies as those in the benchmark test, we were able to find novel mappings that could have not been derived from the existing ontologies.

2.3 Error Analysis

We have performed an error analysis in order to identify possible ways in which the alignment's precision could be improved. Table 3 shows the various types of false mappings and their numbers. We have identified four types of false mappings. First, we found mappings that simply stated a false statement about the domain and which were derived from ontologies containing such incorrect domain knowledge (e.g., $Person \sqsupseteq Event$). Another class of false errors were derived due to incorrect anchoring of the source concepts into the online domain ontologies. For example, the mapping $Academic \sqsupseteq Lecturer$ is false, because in the context of the source ontology $Academic$ refers to academic publications and not to a type of employees.

The largest set of false mappings were due to relations derived by inheritance from high-level, generic concepts such as $Thing$. For example, we established a mapping called $editorBook$ between $Book$ and $Report$ because in one ontology[3] the following path has been followed:

[3] http://www.aifb.uni-karlsruhe.de/WBS/meh/mapping/data/swrc1a.rdf

Test Set	False	False anchor	False generic-c	False part of	Total False
301	4	4	33	-	41
302	15	5	32	-	52
303	81	23	118	3	225
304	77	15	36	3	131

Table 3. Error analysis.

$Report \sqsubseteq Publication \sqsubseteq Root$ and $editorBook(Book, Root)$

Indeed, due to the particular modeling followed by the swrc1a.rdf ontology, it has lead to 75 out of 118 false mappings in this category. These were mostly caused by properties which had $Root$ as a domain or range and which were then inherited by the subclasses of $Root$.

Finally, some subsumption mappings were established between concepts that are in fact related by meronymy relations (e.g., $Journal \sqsupseteq Article$).

3 General comments

3.1 Comments on the results

The results obtained by our in-house evaluation show that it is possible to obtain alignments containing not only equivalent mappings and that the precision of the non-equivalence mappings is around 60% if we take into account redundant mappings and 50% when the redundant mappings are excluded. This results are encouraging and could be further improved as discussed in the next section.

3.2 Discussions on the way to improve the proposed system

Our current efforts focus on automatic ways for filtering out a significant part of the incorrect mappings. First, we are currently finalizing a more complex anchoring mechanism for Scarlet which goes beyond lexical comparison of strings. An initial feasibility study of improving anchoring has been presented in [1]. Second, some of the heuristics we observed could be used to build filters for excluding potentially false mappings - e.g., mappings relying on very long inheritance paths and/or containing generic concepts such as $Root, Thing, Agent$, etc.

Now that we have a better insight in the additions one can bring to the alignment based on equivalence mappings only, we will consider building a hybrid matcher which better integrates CIDER and Scarlet instead of just running them sequentially. For example, we wish to include the process of checking whether a mapping is redundant or not within the matching process itself (and not just running it on the final alignment).

3.3 Comments on the OAEI 2008 procedure

Our main conclusion related to OAEI is that it would be beneficial to extend it with support for evaluating non-equivalence mappings as well, possibly for all test cases.

The evaluation of alignments, in general, is a difficult task, with many open questions persisting even in the case of equivalence mappings. Non-equivalence mappings introduce an extra level of complexity as, unlike in the case of equivalence mappings, it is difficult (if not impossible) to manually build reference alignments for such cases. Therefore the automatic assessment of such mappings by following the model used for equivalence mappings is not feasible.

An interesting line of work described in [4] and [7] is to use logical reasoning in order to assess the quality of mappings in a given alignment. Their assumption is that mappings which introduce logical inconsistencies are likely to be incorrect and should be eliminated. We think that this work could be used for automatically assessing some of the non-equivalence mappings.

One of the shortcoming of the above mentioned methods is that they are hampered by underspecified ontologies. Also, so far, they are only able to assess the quality of subsumption and equivalence mappings and have not considered disjoint and generic, named relations.

To address these problems we envision the development of a set of methods which rely on a different paradigm. Namely, they would use the Web (or other knowledge sources, e.g., Wikipedia, online ontologies) to predict the correctness of a given mapping automatically. For example, in their recent paper [3], Gracia and Mena have shown that web-based relatedness measures can judge the correctness of a mapping almost as well as humans do. Their measure reached the same conclusion as human evaluators for 80% of a corpus of 160 mappings. This is a remarkable result given the fact that inter-evaluator agreement between humans is often as low as 70%. While the results of such evaluation might be slightly less precise than human evaluation, a key advantage is that all submitted alignments would be judged in a uniform and robust way.

3.4 Proposed new measures

Based on the lessons from our evaluation, we think that making a clear distinction between redundant and non-redundant mappings is an important issue and also a process that can be easily automated. According to this we have devised two precision values.

Having said that, we think that the measures will highly depend on the concrete evaluation procedure that will be used, so the measures we presented here might not be feasible in combination with an automated evaluation.

4 Conclusion

We have investigated two main issues related to non-equivalence mappings. First, we have shown that our system can bring an important number of non-redundant and correct non-equivalence mappings to an equivalence based alignment. Our error analysis has also shown that more can be done in order to filter out obviously false mappings.

Second, we have pointed out that, the current OAEI procedure is biased towards dealing with equivalence mappings and as such there is no suitable support for evaluating non-equivalence mappings (except the manual evaluations offered by some of the tests). We think that as the field evolves towards more complex mappings this needs to be taken into account by OAEI. As a first step, we think it could be useful to investigate a set of methods that could be used for automatic mapping evaluation.

Acknowledgements

This work was funded by the NeOn IST-FF6-027595 and CICYT TIN2007-68091-C02-02 projects.

References

1. J. Gracia, V. Lopez, M. d'Aquin, M. Sabou, E. Motta, and E. Mena. Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching. ISWC Workshop on Ontology Matching, 2007.
2. J. Gracia and E. Mena. Ontology Matching with CIDER: Evaluation Report for the OAEI 2008. ISWC Workshop on Ontology Matching, 2008.
3. J. Gracia and E. Mena. Web-based Measure of Semantic Relatedness. In *Proc. of 9th International Conference on Web Information Systems Engineering (WISE 2008), Auckland (New Zealand)*, volume 5175, pages 136–150. Springer Verlag LNCS, ISSN 0302-9743, ISBN 978-3-540-85480-7, September 2008.
4. C. Meilicke, H. Stuckenschmidt, and A. Tamilin. Reasoning Support for Mapping Revision. *Journal of Logic and Computation*, 2008.
5. M. Sabou, M. d'Aquin, and E. Motta. Exploring the Semantic Web as Background Knowledge for Ontology Matching. *Journal on Data Semantics*, XI, 2008.
6. R. Trillo, J. Gracia, M. Espinoza, and E. Mena. Discovering the Semantics of User Keywords. *Journal on Universal Computer Science. Special Issue: Ontologies and their Applications*, November 2007.
7. P. Wang and B. Xu. Debugging Ontology Mappings: A Static Approach. *Computing and Informatics*, 27(1), 2008.

TaxoMap in the OAEI 2008 alignment contest

Fayçal Hamdi[1], Haïfa Zargayouna[2], Brigitte Safar[1], and Chantal Reynaud[1]

[1] LRI, Universit Paris-Sud, Bt. G, INRIA Futurs
2-4 rue Jacques Monod, F-91893 Orsay, France
firstname.lastname@lri.fr
[2] LIPN, Université Paris 13 - CNRS UMR 7030,
99 av. J.B. Clément, 93440 Villetaneuse, France.
haifa.zargayouna@lipn.univ-paris13.fr

Abstract. TaxoMap is an alignment tool which aim is to discover rich correspondences between concepts. It performs an oriented alignment (from a source to a target ontology) and takes into account labels and sub-class descriptions. Our participation in last year edition of the competition have put the emphasis on certain limits. TaxoMap 2 is a new implementation of TaxoMap that reduces significantly runtime and enables parameterization by specifying the ontology language and different thresholds used to extract different mapping relations. The new implementation stresses on terminological techniques, it takes into account synonymy, and multi-label description of concepts. Special effort was made to handle large-scale ontologies by partitioning input ontologies into modules to align. We conclude the paper by pointing out the necessary improvements that need to be made.

1 Introduction

TaxoMap was designed to retrieve useful alignments for information integration between different sources. The alignment process is then **oriented** from ontologies that describe external ressources (named *source* ontology) to the ontology (named *target* ontology) of a web portal. The target ontology is supposed to be well-structured whereas source ontology can be a flat list of concepts.
TaxoMap makes the assumption that most semantic resources are based essentially on classification structures. This assumption is confirmed by large scale ontologies which contain rich lexical information and hierarchical specification without describing specific properties or instances.

To find mappings in this context, we can only use the following available elements: labels of concepts and hierarchical structures.

Previous participation of TaxoMap in the alignment contest [2], despite positive outcome, have put the emphasis on certain limits:

- Multi-label concepts: previous version of TaxoMap assumed that a concept has only one label. This leads to loose interesting relations between multi-label concepts.
- Large ontologies: TaxoMap were unable to run on real ontologies, such as Agrovoc[3].

[3] http://www4.fao.org/agrovoc/

TaxoMap 2 is a new implementation of TaxoMap which aims to overcome these limits and provides modular code (easily extensible). It introduces a morphosyntactic analysis and new heuristics. Moreover, we propose new methods to partition large ontologies into modules which TaxoMap can handles easily.

We take part to four tests. Results on benchmarks are almost the same as last year as the philosophy behind TaxoMap reminds the same (oriented alignment, between concepts only). We perform better -in terms of number of mappings generated and runtime- on Anatomy. Library test allows us to perform a new algorithm for ontology partitioning and to experiments our system with a new language (Dutch). Directory test enables to test our alignment tool in real world taxonomy integration scenario.

2 Presentation of the System

2.1 State, Purpose and General Statement

We consider an ontology as a pair (C, H_C) consisting of a set of concepts C arranged in a subsumption hierarchy H_C. A concept c is defined by two elements: a set of labels and subclass relationships. The labels are terms that describe entities in natural language and which can be an expression composed of several words. A subclass relationship establishes links with other concepts.

Our alignment process is oriented; from a source (O_{Source}) to a target (O_{Target}) ontology. It aims at finding one-to-many mappings between single concepts and establishing three types of relationships, equivalence, subclass and semantically related relationships defined as follows.

Equivalence relationships An equivalence relationship, *isEq*, is a link between a concept in O_{Source} and a concept in O_{Target} with labels assumed to be similar.

Subclass relationships Subclass relationships are usual *isA* class links. When a concept c_S of O_{Source} is linked to a concept c_T of O_{Target} with such a relationship, c_T is considered as a super concept of c_S.

Semantically related relationships A semantically related relationship, *isClose*, is a link between concepts that are considered as related but without a specific typing of the relation.

2.2 Techniques Used

TaxoMap 2 improves terminology alignment techniques. The use of TreeTagger [3], a tool for tagging text with part-of-speech and lemma information, enables to take into account the language, lemma and an use word categories in an efficient way. TaxoMap performs a linguistic similarity measure between labels of concepts. The measure takes into consideration categories of words which compose a label. The words are classified as functional (verbs, adverbs or adjectives) and stop words (articles, pronouns).

Stop words category enables to ignore these words in similarity computation. Functional words has less power than all the other (noun, etc.). The position of a word in the label is also of importance, a common word between two labels is less important after a preposition than a word that is a head. TreeTagger, however, is error-prone, due essentially to short labels.

The main methods used to extract mappings between a concept c_s in O_{Source} and a concept c_t in O_{Target} are:

- Label equivalence: An equivalence relationship, *isEq*, is generated if the similarity between one label of c_s and one label of c_t is greater than a threshold (Equiv.threshold).
- Label inclusion (and its inverse) and hidden inclusion: Then, we consider inclusion of label words: let c_t be the concept in O_{Target} with the highest similarity measure with c_s. If one of the labels of c_t is included in one of the labels of c_s, we propose a subclass relationship $< c_s \ isA \ c_t >$. Inversely, if one of the labels of c_s is included in one of the labels of c_t, we propose a semantically related relationships $< c_s \ isGeneral \ c_t >$. If c_t is not the concept with the highest similarity measure, its measure must be greater than a threshold (HiddenInc.thresholdSim) and the highest similarity measure must be greater than another threshold (HiddenInc.thresholdMax). The intuition behind this strategy is to extract hidden inclusion.
- Reasoning on similarity values : Let c_{tMax} and c_{t2} be the two concepts in O_{Target} with the highest similarity measure with c_s, the relative similarity is the ratio of c_{t2} similarity on c_{tMax} similarity. If the relative similarity is lower than a threshold (isA.threshold), one of the three following techniques can be used:
  - the relationship $< c_s \ isClose \ c_{tMax} >$ is generated if the similarity of c_{tMax} is greater than a threshold (isCloseBefore.thresholdMax) and if one of the labels of c_s is included in one of the labels of c_{tMax}.
  - the relationship $< c_s \ isClose \ c_{tMax} >$ is generated if the similarity of c_{tMax} is greater than a threshold (isClose.thresholdMax).
  - an isA relationship is generated between c_s and the father of c_{tMax} if the similarity of c_{tMax} is greater than a second threshold (isA.thresholdMax).
- Reasoning on structure: an isA relationship is generated if the three concepts in O_{Target} with the highest similarity measure with c_s have similarity greater than a threshold (Struct.threshold), and has a common father.

2.3 Partitioning of large scale ontologies

We propose two methods of ontology partitioning. The aim of our methods is to have minimum blocs to align with maximal number of concepts (that TaxoMap is able to handle). The originality of our methods is that they are *alignment oriented*, that means that the partitioning process is influenced by the mapping process.

The two methods relies on the implementation of PBM[4] algorithm. PBM partitions large ontologies into small blocks (or modules) and construct mappings between the blocks, using predefined matched class pairs, called *anchors* to identify related blocks. We only reuse the partitioning part and the idea of anchors, but adapt them in order to take into account the alignment process in the partitionning. We identify the set of *anchors* as the set of concepts which have the same label in the two ontologies. Even on very large ontologies, this set is computable with a fast and strict equality measure. We also used the possible dissymmetry between ontologies to order the partitionning: if one ontology is well-structured, it will be easier to split it up into cohesive modules, and its partitionning can be used as guideline to partition the other ontology.

The methods proposed are as follows:

- Method1 (see figure 1):
  1. Use PBM algorithm to partition the target ontology O_T into some blocs B_{Ti}.
  2. Identify the set of anchors included in each module B_{Ti}. This set will be the kernel or *center* CB_{Si} of the future module B_{Si} which will be generated from the source ontology O_S.
  3. Use PBM algorithm to partition the source ontology around the identified centers CB_{Si}.
  4. Align each module B_{Si} with the corresponding module B_{Ti}.
- Method2 (see figure 2):
  1. Partition the target ontology O_T by modifying PBM algorithm in order to take into account anchors. Generated modules contain coherent set of concepts that maximize the number of anchors.
  2. Partition the source ontology O_S the same way then step 1. The interesting anchors that influence partitioning are those that goes in the same module generated from O_T.
  3. Align modules that share maximal number of anchors.

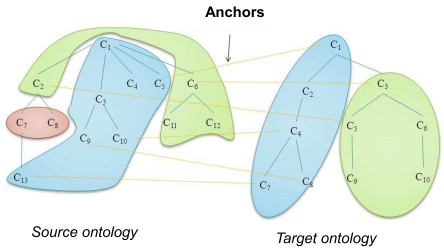


Fig. 1. *Method1 for partitioning*

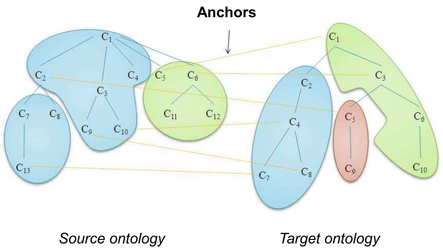


Fig. 2. *Method2 for partitioning*

2.4 Adaptations made for the Evaluation

We do not make any specific adaptation in the OAEI 2008 campaign. All the alignments outputted by TaxoMap are uniformly based on the same parameters. For library test, the language was set to *nl* (for Dutch). We had, however, fixed confidence values depending on relation types.

2.5 Link to the system and parameters file

TaxoMap requires :

- Mysql
- Java (from 1.5)
- TreeTagger [4] with its language parameter files.

The version of TaxoMap (with parameter files) used in 2008 contest can be downloaded from:

- http://www.lri.fr/˜haifa/TaxoMap.jar: a parameter lg has to be specified it denotes the language of the ontology. For example TaxoMap.jar *fr* to perform alignment on ontologies in French. If no language is specified, it is supposed to be English.
- http://www.lri.fr/˜haifa/TaxoMap.properties: a parameter file which specifies:
  - The command to launch tree-tagger.
  - Treetagger word categories that has to be considered as functional, stop words and prepositions.
  - The RDF output file.
  - Different thresholds of similarity, depending on the method used.
- http://www.lri.fr/˜haifa/dbproperties.properties: a parameter file which contains the user and password to access to MySql.

2.6 Link to the Set of Provided Alignments

The alignments produced by TaxoMap are available at the following URLs:
http://www.lri.fr/˜haifa/benchmarks/
http://www.lri.fr/˜haifa/anatomy/
http://www.lri.fr/˜haifa/directory/
http://www.lri.fr/˜haifa/library/

3 Results

3.1 Benchmark Tests

Since our algorithm only considers labels and hierarchical relations and only provides mapping for concepts, the recall is low even for the reference alignment. The overall results are almost similar -with no surprise- to those of last year.

The whole process of alignment costs less than 2 minutes.

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

3.2 Anatomy Test

The anatomy real world case is to match the Adult Mouse Anatomy (denoted by *Mouse*) and the NCI Thesaurus describing the human anatomy (tagged as *Human*). *Mouse* has 2,744 classes, while *Human* has 3,044 classes. As last year, we considered *Human* as the target ontology as is it well structured and larger than *Mouse*.
TaxoMap gains considerably on runtime, it performs the alignment (with no need to partition) in about 25 minutes which is better than last year where TaxoMap took about 5 hours to align the two ontologies.

TaxoMap generates much more mappings than last year. Only about 200 concepts were left unmapped, whereas last year it was nearly 900.
As only equivalence relationships will be evaluated, we change different mapping types to equivalence with these confidence values:

- (type1) For *isEq* and *isClose* relations, confidence value was set to 1.
- (type2) For *isA* relations generated by label inclusion, confidence value was set to 0.8.
- (type3) For *isA* relations generated by structural technique or by relative similarity method, confidence value was set to 0.5.

TaxoMap discovers 2 533 mappings: 1 208 type1 relations, 1 190 type2 relations and 135 type3 relations. The improvement in comparaison with last year results relies on the use of TreeTagger and on taking into account synonymy.

3.3 Directory Test

The directory task consists of Web sites directories like Google, Yahoo! or Looksmart. To date, it includes 4,639 tests represented by pairs of OWL ontologies. TaxoMap takes about 40 minutes to complete all the tests.

3.4 Library Test

The library task includes two SKOS thesauri GTT and Brinkman thesauri. Since TaxoMap focuses on Web ontologies expressed in RDFS and OWL, we have to adopt two OWL version ontologies transformed by campaign organizers in this task. GTT owns 35,000 classes, while Brinkman thesauri owns 5,000 classes. The main drawback of using OWL ontologies is that there is no distinction in OWL descriptions (rdfs:label statements) between skos:prefLabel, skos:altLabel and skos:hiddenLabel statements, which removes the subtle distinctions that exist between these different properties.

We applied the first method of partitioning, this is due to the fact that only 3535 anchors were discovered and that the two ontologies were poorly structured. As the method2 relies on these two informations simultaneously, the partitioning results were not judged relevant.

The partitioning of Brinkman thesauri (considered as target ontology) leads to 227 modules, the largest module contains 703 concepts. GTT (source ontology) is partitionned into 18 306 modules, 16 265 modules contain only one concept, the largest module contains 517 concepts. We performed 212 combinations that leads to 3 217 mappings.

The fact that the total number of mappings is less than the number of found anchors is due to the fact that anchors are computed between labels (a concept described by three labels can have three anchors, which is not the case for mappings, where a concept is matched to only one concept). As alignments are performed between modules, this can lead to loose some potential mappings. This is particularly the case of all modules that contain only one concept, as they are ignored by the alignment process.

As skos relations will be evaluated, we change different mapping types to skos ones with these confidence values:

- (type1) *isEq* relations become skos:exactMatch with a confidence value set to 1.
- (type2) *isA* relations become skos:narrowMatch with a confidence value set to 1 for label inclusion, 0.5 for relations generated by structural technique or by relative similarity method.
- (type3) *isGeneral* relations become skos:broadMatch with a confidence value set to 1.
- (type4) *isClose* relations become skos:relatedMatch with a confidence value set to 1.

Generated mappings are as follows: 1 872 type1 relations, 1 031 type2 relations, 274 type3 relations and 40 type4 relations. The whole process of alignment costs about 40 minutes. The partitioning process costs nearly 2 hours.

The language of both thesauri is Dutch, we launched tree-tagger with Dutch parameter file. The main difficulty is that there were no Tagset description given for this language and it was difficult to specify word categories needed for the linguistic similarity method.

4 General Comments

4.1 Results

TaxoMap 2 significantly improves the results on the previous version of TaxoMap in terms of runtime and number of generated mappings. The new implementation offers extensibility and modularity of code. TaxoMap can be parameterized by the language used in ontologies and different thresholds. We put the emphasis on terminological alignment by taking into account synonymy and multi-label concepts. Our partitioning algorithms allows us to participate to tests with large ontologies.

4.2 Future Improvements

The following improvements can be made to obtain better results:

- Use of WordNet as a dictionary of synonymy. The synsets can enrich the terminological alignment process if an *a priori* disambiguation is made.
- To develop the remaining structural techniques which proved to be efficient in last experiments [5] [6].

5 Conclusion

This paper reports our participation to OAEI campaign with a new implementation of TaxoMap. Our algorithm proposes an oriented mapping between concepts. TaxoMap 2 is better now than last year. Due to partitioning, it is able to perform alignment on real-world ontologies. Our participation in the campaign allows us to test the robustness of TaxoMap, our partitioning algorithms and new terminological heuristics.

References

[1] Lin, D. : An Information-Theoretic Definition of Similarity. ICML. Madison. (1998) 296–304

[2] Zargayouna, H., Safar, B., and Reynaud, C. : TaxoMap in the OAEI 2007 alignment contest Proceedings of the ISWC'07 workshop on Ontology Matching OM-07 (2007) 268-275

[3] Schmid H. : Probabilistic Part-of-Speech Tagging Using Decision Trees. International Conference on New Methods in Language Processing (1994)

[4] Hu, W., Zhao, Y., and Qu, Y. Partition-based block matching of large class hierarchies. Proc. of the 1st Asian Semantic Web Conference (ASWC06). (2006) 72-83

[5] Reynaud, C., Safar, B. When usual structural alignment techniques don't apply The ISWC'06 workshop on Ontology matching (OM-06). (2006)

[6] Reynaud, C., Safar, B. Exploiting WordNet as Background Knowledge The ISWC'07 Ontology Matching (OM-07). (2007)

On Applying Matching Tools to Large-Scale Ontologies

Heiko Paulheim

SAP Research
heiko.paulheim@sap.com

Abstract. Many existing ontology matching tools are not well scalable. In this paper, we present the *Malasco* system, which serves as a framework for reusing existing, non-scalable matching systems on large-scale ontologies. The results achieved with different combinations of partitioning and matching tools are discussed, and optimization techniques are examined. It is shown that the loss of result quality when matching with partitioned data can be reduced to less than 5% compared to matching with unpartitioned data.

1 Introduction

The need for matching large-scale ontologies arises in different fields. In electronic business, several large ontologies representing business standards are in use [1]. Another example is the field of medical research where large databases and ontologies exist in which taxonomies, definitions, and experimental results are stored.

To the best of our knowledge, there are only three works which explicitly address the scalability issue: The schema matching system COMA++ [2], the ontology matching tool Falcon-AO [3], and the MOM approach [4]; however, there seems to be no publicly available implementation of the latter. All of them address the scalability problem by first partitioning the input ontologies into smaller sub-ontologies and then performing the actual matching task on the partitions. This approach seems promising, although one must take care to implement the partitioning step in a way that large ontologies can be processed, in order not to replace one bottleneck with another.

2 Scalability of Existing Matching Tools

To examine the scalability of existing ontology matching tools, we used two pairs of large ontologies: the e-business standards *eClass* (375K triples) and *UNSPSC* (83K triples), and the medical ontologies *GO* (465K triples) and *NCI thesaurus* (543K triples). From the large variety of matching tools, we chose tools that are publicly available and widely known, two of which focus explicitly on the matching of large-scale ontologies. We conducted tests with the above mentioned

COMA++ and Falcon-AO, as well as FOAM, INRIA, PROMPT, and CROSI (using a simple string comparator), on a standard desktop PC.

The business pair of only be processed by COMA++ and Falcon-AO. The larger medical pair could not be processed by any of the tools examined. Most of the tools suffered from a lack of memory. These experiments show that matching large ontologies is a severe problem with many of the tools that are currently available.

3 The Malasco System

The system introduced in this paper is called *Malasco* (**Ma**tching **la**rge **sc**ale **o**ntologies). It allows matching large-scale ontologies by first partitioning the input ontologies. The actual matching is then carried out on the smaller partitions.

3.1 Design

This approach has also been implemented in COMA++ and Falcon-AO. Unlike those systems, our implementation follows a more modular design, which allows the use of different existing systems both for partitioning and for matching the partitions. This approach has several advantages:

- Existing matching and partioning tools can easily be reused. This lowers the effort of setting up a matching solution and offers the possibility to benefit from future developments without having to modify the system.
- Different matching tools provide results of various quality, depending on the nature of the input ontologies. Therefore, building a system that can work with different matching tools is a promising approach for creating a versatile tool.
- From an academical point of view, the approach allows experiments on different combinations of partitioning and matching tools.

3.2 Partitioning approaches

As a simple partitioning approach, we implemented a naive baseline algorithm which iterates over the RDF sentences [5] and creates chunks of N triples. While that approach is rather naive (as it does not create clusters of concepts that are semantically related), two more sophisticated algorithms are used in the prototype: the islands algorithm developed by Stuckenschmidt and Klein [6], implemented in the tool *PATO* and the ε-connections algorithm proposed by Grau et al. [7], implemented in the tool *SWOOP*.

4 Evaluation

The Malasco system has been evaluated in two respects: the ability to process large-scale ontologies, and the quality of the matching results achieved.

4.1 Scalability

To demonstrate that our system is capable of matching large-scale ontologies, we used the test ontologies and test environment described in section 2. As an example, we used the baseline algorithm with a maximum partition size of 5,000 statements and the INRIA matching system. Our system could process both pairs; the largest amount of time – more than 100 times longer than the rest of the process – was consumed by the pairwise matching of partitions.

4.2 Result Quality

While it is obvious that element-based matching algorithms can be run on partitions of the input ontologies with unchanging results (given a covering partitioning), most matching systems are structure-based and will thus produce different (and probably worse) results on partitioned data. To evaluate how big the loss of quality is when working on partitioned data, we ran the example matching tools both on unpartitioned and on partitioned ontologies and compared the results. Since the matching tools examined can only work on smaller-scale ontologies, such a comparison is only possible on smaller-scale data sets.

Six pairs of ontologies of a size between 600 and 2,000 statements were used for evaluation. For partitioning, we used two variants of the baseline algorithm (with 250 and 500 statements as a maximum), two variants of the islands algorithm (with 50 and 100 classes per island as a maximum), the ε-connections algorithm[1], and the unpartitioned ontologies for comparison. For pairwise matching of the partitions, we used INRIA [8] and FOAM [9] in their respective standard configurations (both of which partly rely on structure-based algorithms).

To evaluate the results, we calculated recall, precision, and F-measure. While the recall value achieved on partitioned data is as high as (and in some cases even slightly higher than) the result on unpartitioned data, the precision value is less than 50% than that achieved on unpartitioned data, caused by a very high number of false positives.

4.3 Optimization I: Using overlapping partitions

To achieve better results, in particular better precision values, we tested two optimization approaches. The first one is motivated by the insight that structure-based matching approaches use information on neighboring elements. For partioned ontologies, those are missing for elements on the border of a partition. Hence, for the first optimization approach, we added the direct neighbors for each concept contained in a partition, thus creating overlapping partitions. The matching is then performed on the overlapping partitions. Mapping elements found between the neighboring elements are discarded, because the matching algorithm only has partial information about those elements.

When using overlapping partitions, it can be observed that using overlapping partitions causes a significant improvement of the precision value (the loss

[1] For the ε-connections algorithm, various problems can be observed [7]; two ontologies could not be partitioned at all. Therefore, that algorithm is considered not suitable and not regarded any further in the following results.

of precision can be limited to less than 20%), almost without any negative affection of the recall value. On the other hand, since the overlapping partitions are larger, the matching phase runs up to four times as long as for non-overlapping partitions.

4.4 Optimization II: Thresholding

The second approach to improve our system's results' quality is the use of a lower threshold. As most matching system provide a confidence parameter with each mapping element, a lower threshold can be employed to discard all elements with a confidence value below that threshold in order to improve the results [10]. This approach has been motivated by the observation that the average confidence value is significantly lower for false positives than for true positives.

To determine an optimal lower threshold τ, we calculated precision, recall, and F-measure for threshold values between 0 and 1 and determined the average optimal (w.r.t. F-measure) threshold values for each partitioning algorithm, including the unpartitioned case for comparison.

Thresholding the results leads to a significant improvements in precision and F-measure. Fig. 1 shows the results using the matching system FOAM[2]. The improvement is stronger than using overlapping partitions, more than 95% of the F-measure achieved on unpartitioned data can be reached (even up to 99% for INRIA). Applying a filter which is optimal for a given partitioning technique leads to almost the same results, thus, the choice for an actual partitioning algorithm is of marginal effect.

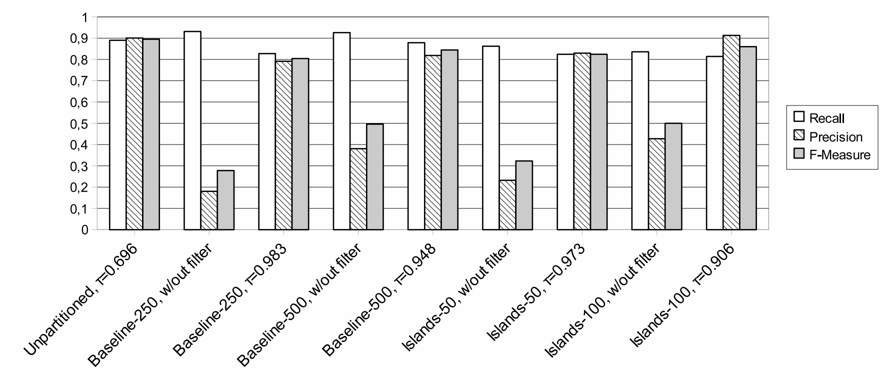


Fig. 1. Results with FOAM and thresholding

Using the thresholding optimization is less costly than using overlapping partitions: the matching system does not have to work on larger partitions, and the runtime complexity of applying the threshold is only linear in the number of results. Combining both overlapping partitions and thresholding leads only to

[2] The results with INRIA were in most of the cases comparable to those achieved with FOAM and are therefore not shown separately.

minimal improvements (less than 5%) compared to thresholding alone. Since using overlapping partitions is rather costly, thresholding alone is the more approriate approach in most usage scenarios.

5 Conclusion

In this paper, we have presented the Malasco framework which allows using existing matching tools for matching large-scale ontologies. Its modular architecture allows for using arbitrary partitioning and matching tools, including domain-specific tools for particular matching tasks.

In our evaluation, we have shown that our system is actually capable of matching large ontologies, that the choice of a particular partitioning algorithm is only of minor importance, and that the quality deviation compared to the results which would be achieved on the unpartitioned ontologies (given a matcher that could process them) can be reduced to less than 5%.

Acknowledgements

The author would like to thank the members of the e-business integration group at Hochschule Darmstadt and the members of the knowledge engineering group at TU Darmstadt for their valuable and helpful remarks.

References

1. Rebstock, M., Fengel, J., Paulheim, H.: Ontologies-based Business Integration. Springer (2008)
2. Do, H.H., Rahm, E.: Matching large schemas: Approaches and evaluation. Information Systems **32**(6) (2007) 857–885
3. Hu, W., Zhao, Y., Qu, Y.: Partition-based block matching of large class hierarchies. [11]
4. Wang, Z., Wang, Y., Zhang, S., Shen, G., Du, T.: Matching large scale ontology effectively. [11]
5. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on rdf sentence graph. In Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J., eds.: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, ACM (2007) 707–716
6. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. [12] 289–303
7. Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A.: Modularizing OWL ontologies. In Sleeman, D., Alani, H., Brewster, C., Noy, N., eds.: Proceedings of the KCAP-2005 Workshop on Ontology Management. (2005)
8. Euzenat, J.: An API for ontology alignment. [12] 698–712
9. Ehrig, M.: Ontology Alignment - Bridging the Semantic Gap. Semantic Web and Beyond. Computing for Human Experience. Springer, New York (2007)
10. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Berlin, Heidelberg, New York (2007)
11. Mizoguchi, R., Shi, Z., Giunchiglia, F., eds.: The Semantic Web - ASWC 2006, First Asian Semantic Web Conference. Number 4183 in LNCS, Springer (2006)
12. McIlraith, S.A., Plexousakis, D., van Harmelen, F., eds.: The Semantic Web - ISWC 2004: Proceedings of the Third International Semantic Web Conference. Number 3298 in LNCS, Springer (2004)

Literature-based alignment of ontologies

Patrick Lambrix and He Tan and Wei Xu

Department of Computer and Information Science
Linköpings universitet, Sweden

Abstract. In this paper we propose and evaluate new strategies for aligning ontologies based on text categorization of literature using support vector machines-based text classifiers, and compare them with existing literature-based strategies. We also compare and combine these strategies with linguistic strategies.

1 Introduction

In recent years many ontologies have been developed and many of these ontologies contain overlapping information. A number of ontology alignment systems that support the user to find inter-ontology relationships exist (see overviews in e.g., [2, 5] and http://www.ontologymatching.org/). Recently, there is a growing interest in instance-based methods for ontology alignment. In this paper we slightly generalize the method for instance-based ontology alignment using literature that was proposed in [7]. Further, we propose a new instantiation of the method based on text categorization using support vector machines (SVMs). We evaluate these algorithms in terms of the quality of the alignment results for the five test cases used in [7]. We compare two SVM-based algorithms with each other and with the Naive Bayes text classification approach of [7]. Finally, we compare the algorithms with a good text-based approach and discuss the advantages and disadvantages of combining the approaches. For related work, more results and more details we refer to the longer version of this paper that is available from the SAMBO website (http://www.ida.liu.se/∼iislab/projects/SAMBO/).

2 Background

Many ontology alignment systems are based on the computation of similarity values between terms in different ontologies and can be described as instantiations of the general framework defined in [5]. An alignment algorithm receives as input two source ontologies. The algorithm can include several matchers. These matchers calculate similarities between the terms from the ontologies. Alignment suggestions are then determined by combining and filtering the results generated by one or more matchers. The suggestions are then presented to the user who accepts or rejects them.

A method for creating a matcher that uses scientific literature was proposed in [7]. It builds on the intuition that a similarity measure between concepts can be computed based on relationships between the documents in which they are used. It contains the following basic steps (slightly generalized). (1) **Generate corpora.** For each ontology that we want to align we generate a corpus of documents. (2) **Generating classifiers.** For each ontology one or more document classifiers are generated. The corpus of documents associated to an ontology is used for generating its related classifiers. (3) **Classification.** Documents of one ontology are classified by the document classifiers of the other ontology and vice versa. (4) **Calculate similarities.** A similarity measure between concepts in the different ontologies is computed based on the results of the classification.

In [7] an instantiation (NB) of this method was implemented and evaluated using test cases involving biomedical ontologies. For step 1 a corpus was generated by querying PubMed (October 23, 2005) with each concept and retrieving the 100 most recent abstracts (if there were so many) for each concept. In step 2 one Naive Bayes classifier per ontology was generated. The classifiers return for a given document d the concept C in the ontology for which the posterior probability $P(C|d)$ results in the highest value. In step 3 the Naive Bayes classifier for one ontology was applied to every abstract in the abstract corpus of the other ontology and vice versa. Finally, in step 4 a similarity value between two concepts was computed using the numbers of abstracts associated with one concept that are also related to the other concept as found by the classifiers.

In general, in step 2 a document may be assigned to several concepts and thus we may regard the classification of documents to concepts as several binary classification problems, one for each concept in an ontology. In the next section we propose an instantiation of the method that does exactly this and is based on SVMs. SVMs [8] is a machine learning method that constructs a separating hyperplane in a feature space between two data sets (positive and negative examples) which maximizes the margin between the two sets. The setting can also be generalized to learning from positive and unlabeled examples (e.g. [6]).

3 Alignment algorithms

The basic algorithm implements the steps as follows. (1) **Generate corpora.** We used the same corpora as in [7]. (2) **Generating the classifiers.** For each concept in each ontology an SVM text classifier was generated. We used the LPU [6] system. LPU generates text classifiers based on positive and unlabeled examples. The abstracts retrieved when querying for a concept were used as positive examples for that concept. Further, for a given concept we used one abstract of each other concept in the same ontology as unlabeled examples. The SVM text classifier for a concept returns for a given document whether the document is related to the concept. It returns a value that is positive if the document is classified to the concept and negative otherwise. (3) **Classification.** The SVM text classifier for each concept in one ontology is applied to every abstract in the abstract corpus of the other ontology and vice versa. The classification was done by using the text classifiers generated by LPU within the SVM^{light} system [4]. Observe that a document can be classified to zero, one or more than one concept in an ontology. (4) **Calculate similarities.** We define the similarity between a concept C_1 from the first ontology and a concept C_2 from the second ontology as:

$$\frac{n_{SVMC-C_2}(C_1, C_2) + n_{SVMC-C_1}(C_2, C_1)}{n_D(C_1) + n_D(C_2)}$$

where $n_D(C)$ is the number of abstracts originally associated with C, and $n_{SVMC-C_q}(C_p, C_q)$ is the number of abstracts associated with C_p that are also related to C_q as found by classifier $SVMC - C_q$ related to concept C_q.

The pairs of concepts with a similarity measure greater or equal than a predefined threshold are then presented to the user as candidate alignments.

In NB a document was classified to exactly one concept. We wanted to evaluate whether this has a real influence in the similarity computation. Therefore, we also developed an alternative to the basic SVM algorithm where in step 3 a document can be classified to only one concept. We assign a document only to the concept for which its SVM classifier generated the highest positive value for that document. In the case more than one classifier produces the highest positive value, then one of the associated concepts is chosen.

4 Evaluation

We evaluate the proposed algorithms with respect to the quality of the suggestions they generate. We also compare them to NB as well as to the best text-based matcher (TermWN) implemented in SAMBO [5]. Further, we investigate the combination of the proposed algorithms and TermWN.

We used the following set-up. We use the same five **test cases** as in [7]. For the first two cases we use a part of a Gene Ontology (GO) ontology together with a part of Signal Ontology (SigO). The first case, *B* (behavior), contains 57 terms from GO and 10 terms from SigO. The second case, *ID* (immune defense), contains 73 terms from GO and 17 terms from SigO. The other cases are taken from the anatomy category of Medical Subject Headings (MeSH) and the Adult Mouse Anatomy (MA): *nose* (containing 15 terms from MeSH and 18 terms from MA), *ear* (containing 39 terms from MeSH and 77 terms from MA), and *eye* (containing 45 terms from MeSH and 112 terms from MA). Golden standards for these cases were developed by domain experts. Further, we use the same **corpus** as in [7]. We use SVM-based **matchers** based on sets of maximum 100 documents per concept. These matchers are denoted as SVM-P and SVM-S where P and S stand for Plural (a document can be classified to several concepts) and Single (a document can be classified to only one concept), respectively.

The results are given in table 1. The first column represents the cases and the number of expected alignments for each case based on the golden standards. The expected alignments are a minimal set of suggestions that matchers are expected to generate for a perfect recall. The second column represents threshold values. The cells in the other columns contain quadruplets a/b/c/d which represent the number of a) suggestions, b) correct suggestions, c) wrong suggestions and d) inferred suggestions, for a given case, matcher and threshold.

Comparison of single and plural assignment. The recall for the plural assignment is much higher than the recall for the single assignment. This comes, however, at a cost. The precision for the single assignment algorithm is much higher than for the plural assignment algorithm. We see a real trade-off here: find many expected alignments, but also get many wrong suggestions, or, find few expected alignments, but receive almost no wrong suggestions.

Comparison of NB and SVM-S. These two single assignment algorithms give relatively few suggestions but have high precision. However, NB gives always more suggestions than SVM for the same threshold. NB also always gives suggestions, except for case ID and threshold 0.8, while SVM-S often does not give suggestions. It is clear that SVM-S does not perform well with high thresholds. In general, NB has slightly better recall than SVM-S, while SVM-S has slightly higher precision than NB.

	Th	SVM-P	SVM-S	NB	TermWN	TermWN+	TermWN+
						SVM-S	SVM-P
B	0.4	387/4/258/125	0/0/0/0	4/2/1/1	58/4/22/32	4/4/0/0	156/4/84/68
4	0.5	306/4/203/99	0/0/0/0	2/2/0/0	35/4/13/18	4/4/0/0	52/4/19/29
	0.6	225/4/148/73	0/0/0/0	2/2/0/0	13/4/4/5	0/0/0/0	21/4/7/10
	0.7	130/3/79/48	0/0/0/0	2/2/0/0	6/4/0/2	0/0/0/0	7/4/1/2
	0.8	36/0/22/14	0/0/0/0	1/1/0/0	4/4/0/0	0/0/0/0	4/4/0/0
ID	0.4	672/8/592/72	2/2/0/0	9/6/3/0	96/7/66/23	8/6/2/0	302/8/262/32
8	0.5	490/8/433/28	0/0/0/0	5/5/0/0	49/7/25/17	6/6/0/0	155/7/127/21
	0.6	336/8/300/28	0/0/0/0	2/2/0/0	16/5/5/6	2/2/0/0	71/7/48/16
	0.7	222/6/191/25	0/0/0/0	1/1/0/0	7/5/2/0	1/1/0/0	19/7/7/5
	0.8	108/5/93/10	0/0/0/0	0/0/0/0	6/4/0/2	0/0/0/0	7/5/2/0
nose	0.4	155/7/124/24	5/5/0/0	6/5/1/0	48/7/37/4	9/7/2/0	80/7/66/7
7	0.5	120/7/91/22	4/4/0/0	6/5/1/0	28/7/18/3	7/7/0/0	58/7/47/4
	0.6	85/7/60/18	2/2/0/0	5/5/0/0	8/6/2/0	6/6/0/0	31/7/47/4
	0.7	58/6/45/7	0/0/0/0	5/5/0/0	6/6/0/0	4/4/0/0	11/7/4/0
	0.8	34/6/27/1	0/0/0/0	3/3/0/0	6/6/0/0	1/1/0/0	6/6/0/0
ear	0.4	1224/24/1056/144	14/12/2/0	18/16/2/0	155/26/110/19	34/25/8/1	585/27/481/77
27	0.5	957/23/822/112	11/10/1/0	15/14/1/0	99/26/65/8	27/23/4/0	203/26/146/31
	0.6	696/22/590/84	1/1/0/0	12/11/1/0	47/26/19/2	17/17/0/0	96/24/64/8
	0.7	478/22/392/64	0/0/0/0	11/10/1/0	34/26/8/0	12/12/0/0	55/23/28/4
	0.8	278/21/223/34	0/0/0/0	3/3/0/0	28/25/3/0	1/1/0/0	29/21/6/2
eye	0.4	2055/25/1926/104	7/7/0/0	25/18/7/0	135/26/100/9	28/23/5/0	643/25/568/50
27	0.5	1481/25/1366/90	4/4/0/0	18/17/1/0	74/23/44/7	21/20/1/0	272/25/221/26
	0.6	957/25/860/72	0/0/0/0	14/14/0/0	33/22/10/1	16/16/0/0	138/24/101/13
	0.7	612/24/539/49	0/0/0/0	10/10/0/0	24/21/3/0	7/7/0/0	54/21/27/6
	0.8	344/23/290/31	0/0/0/0	3/3/0/0	22/20/2/0	0/0/0/0	25/21/4/0

Table 1. Results.

Comparison with and combination with other matchers. The table also shows the quality of the suggestions of TermWN (from [5]), and the combinations (sum, equal weight) of TermWN with SVM-P and SVM-S. TermWN has higher recall than SVM-S and NB. It also has better recall than SVM-P for the case ear, but for the other cases the recall is similar. TermWN has better precision than SVM-P, but worse than SVM-S and NB. Almost all expected alignments were found by at least one SVM or NB matcher and threshold at least 0.4. TermWN with threshold 0.4 missed 1 expected alignment for ID, 1 for ear and 1 for eye.

The combination of TermWN and SVM-S gave perfect results for B and thresholds 0.4 and 0.5. Otherwise, when it gave suggestions, the precision was high. For thresholds 0.4 and 0.5, SVM-S worked as a filter on TermWN by removing many wrong suggestions at the cost of no or few correct suggestions. For higher thresholds too many correct suggestions were removed. For most cases and thresholds the combination of TermWN and SVM-P gave better recall than TermWN and SVM-P. The precision of the combination was higher than the precision for SVM-P, but lower than the precision for TermWN. As shown in the longer version of the paper, the precision for the combination could become better than the precision for TermWN by using the double threshold filtering technique of [1] while keeping the recall at the same level for most cases.

5 Conclusion

We have proposed SVM-based algorithms for aligning ontologies using literature. We have shown that there is a trade-off between the single and plural assignment methods regarding precision and recall. Further, SVM-S and NB obtained similar results. The combinations of TermWN with SVM-S and with SVM-P lead to a large gain in precision compared to TermWN and SVM-P, with still a high recall.

References

1. Chen B, Tan H and Lambrix P. 2006. Structure-based filtering for ontology alignment. *Proceedings of the IEEE WETICE Workshop on Semantic Technologies in Collaborative Applications*, pp 364-369.
2. Euzenat J and Shvaiko P. 2007. *Ontology Matching.* Springer.
3. Joachims T. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning*, LNCS 1398, 137-142.
4. Joachims T. 1999. Making Large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, B Schölkopf and C Burges and A Smola (eds), MIT-Press. http://svmlight.joachims.org/
5. Lambrix P and Tan H. 2006. SAMBO - A System for Aligning and Merging Biomedical Ontologies. *Journal of Web Semantics*, 4(3):196-206.
6. Liu B, Dai Y, Li X, Lee WS and Yu Ph. 2003. Building Text Classifiers Using Positive and Unlabeled Examples. *Proceedings of the Third IEEE International Conference on Data Mining*, 179-188. http://www.cs.uic.edu/∼liub/LPU/LPU-download.html
7. Tan H, Jakoniene V, Lambrix P, Aberg J and Shahmehri N. 2006. Alignment of Biomedical Ontologies using Life Science Literature. *Proceedings of the International Workshop on Knowledge Discovery in Life Science Literature*, LNBI 3886, 1-17.
8. Vapnik V. 1995. *The Nature of Statistical Learning Theory.* Springer.

Ontology Mapping via Structural and Instance-Based Similarity Measures

Konstantin Todorov[1] and Peter Geibel[2]

[1]IKW, Universtity of Osnabrück, Albrechstr. 28, 49076 Osnabrück, Germany
[2]TU Berlin, Fakultät IV, Franklinstr. 28/29, 10587 Berlin, Germany

Abstract. The paper presents an overview of a novel procedure for mapping hierarchical ontologies, populated with properly classified text documents. It combines structural and instance-based approaches to reduce the terminological and conceptual ontology heterogeneity. It yields important granularity and instantiation judgments about the inputs and is to be applied to mapping web-directories.

1 Introduction and Initial Setting

Heterogeneity between ontolgies can occur in many forms, not in isolation from one another [5]. We describe our approach to map two hierarchical, tree-structured ontologies designed to categorize text documents (web pages) with respect to their content, by reducing their terminological and conceptual heterogeneity. The paper extends previous work by one of the co-authors [10]. We make use of both intentional and extensional information contained in the input ontologies and combine them in order to establish correspondences between the ontologies concepts. In addition, the proposed procedure yields assertions on the granularity and the extensional richness of one ontology compared to another which will be helpful at assisting the eventual stage of ontology merging.

Definition 1. *A hierarchical ontology is a pair* $O := (C_O, \texttt{is\_a})$*, where* C_O *is a finite set whose elements are called concepts and* $\texttt{is\_a}$ *is a partial order on* C_O *with the following property:*

- there exists exactly one element $A_0 \in C_O$ *such that* $\{B \in C_O | (A_0, B) \in \texttt{is\_a}\} = \emptyset$*,*

- for every element $A \in C_O$*,* $A \neq A_0$*, there exists an unique element* $A' \in C_O$ *such that* $(A, A') \in \texttt{is\_a}$*.*

We will use the documents assigned to a given concept as instances of that concept in order to model it. A given class is assigned the union of the sets of documents assigned to all nodes subsumed by this class. In Figure 1(a), the node $c2$ contains the documents set $\{d1, d2, d3, d4, d5, d6\}$.

Our inputs are two ontologies O_1 and O_2 together with their corresponding sets of documents $D_{O_1} = \{d_1^{O_1}, ..., d_{n_{O_1}}^{O_1}\}$ and $D_{O_2} = \{d_1^{O_2}, ..., d_{n_{O_2}}^{O_2}\}$, where each document is represented as a TF/IDF vector [7]. We assume that O_1 and O_2

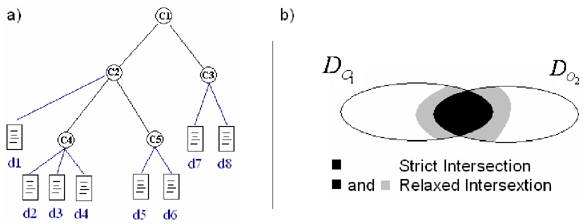


Fig. 1. a: A document populated taxonomy. b: Strict and relaxed intersections.

share a significant extensional overlap and all the documents are in the same natural language.

An entity which plays a key role in our approach is the intersection of D_{O_1} and D_{O_2}. However, when the sets elements are vectors of text documents, it is very likely that the sets contain documents which are similar, but not identical, and therefore not part of the intersection. In order to make use of such documents, we introduce the notion of relaxed intersection (RI) which integrates both identical *and* similar documents from both sets as opposed to the standard strict set intersection (Figure 1(b)): $RI(D_{O_1}, D_{O_2}) = \{d_i^{O_1}, d_j^{O_2} | dist(d_i^{O_1}, d_j^{O_2}) \leq c_d, d_i^{O_1} \in D_{O_1}, d_j^{O_2} \in D_{O_2}\}$, where $dist$ is a properly chosen distance measure on the set of TF/IDF documents [8] and c_d is a similarity parameter to be empirically set. In the sequel, by document set intersection we will mean their relaxed intersection.

2 Structural and Instance-based Mapping Strategies

In the following, we will describe the **structural approach** which forms the first part of our mapping strategy. A hierarchical ontology as described in definition 1 is directly translated to a directed rooted tree $G(V, E)$. Since only hyponimic relations are allowed at this stage, we will assume that the ontology graphs are unlabeled. Bunke et *al.* [1] introduced a graph distance, which accounts for the structural closeness of two taxonomies, represented as non-empty graphs G_1 and G_2:

$$d(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{max(|G_1|, |G_2|)}.$$

The abbreviation mcs stands for maximal common subgraph of G_1 and G_2 defined as a maximal graph isomorphic to sub-graphs of both G_1 and G_2 and $|G|$ denotes the number of vertices in a graph G. The problem of finding a mcs is solved in polynomial time for trees. Various algorithms are discussed in [11].

In addition to the structural approach, we employ **two extensional methods** for deriving concepts similarity assertions. Even though independent from one another, they can be combined yielding an improved similarity learner. In both approaches, we make use of Support Vector Machines (SVMs) [3], operating on the sets of TF/IDF documents assigned to the input ontologies. SVMs are machine learning classifiers which can be trained on a given data set and learn to discriminate between positive and negative instances. For two concepts $A \in C_{O_1}$ and $B \in C_{O_2}$ we define the data sets $S_{O_1} = \{(d_i^{O_1}, y_i^A)\}$ and $S_{O_2} = \{(d_j^{O_2}, y_j^B)\}$, where $d_i^{O_1}, d_j^{O_2} \in \mathbb{R}^d$, $i = 1, ..., n_{O_1}$, $j = 1, ..., n_{O_2}$ with d - the dimension of the TF/IDF vectors. y^A and y^B are labels taking values $+1$ when the corresponding document is assigned to A or B, respectively, and -1 otherwise. The labels separate the documents in each ontology into such that belong to a given concept A or B, respectively (positive instances) and such that do not (negative instances).

One convenient way of making use of extensional information is to model ***concepts as "bags" of instances*** and measure their similarity on set theoretic accounts considering A and B very similar when $A \cap B \approx A$. A standard instance-based similarity measure is the Jaccard coefficient [6], defined as: $Jacc(A, B) = \frac{P(A \cap B)}{P(A \cup B)}$, where $P(X)$ is the probability of a random instance to be an element of X. Note that $P(A \cap B) = P(A, B)$ and $P(A \cup B) = P(A, B) + P(A, \overline{B}) + P(\overline{A}, B)$, where the entity $P(A, B)$ denotes the *joint probability* of A and B. Each of the three joint probabilities is estimated by the fraction of documents that belong to both A and B: $P(A, B) = \frac{|A \cap_{O_1} B| + |A \cap_{O_2} B|}{|D_{O_1}| + |D_{O_2}|}$, where $\cap_{O_1}$ denotes intersecting documents belonging to O_1 only. By training an SVM classifier on the data set S_{O_1} and applying it on the document set D_{O_2} we come up with an estimation of the quantity $|A \cap_{O_2} B|$. Repeating the procedure after inversing the roles of O_1 and O_2 yields $|A \cap_{O_1} B|$. The same algorithm is applied for the other joint probabilities until we have approximations of all of them, as described in [4].

The second extensional indicator for semantic closeness we propose is based on a ***variable selection procedure for SVMs***. Variable selection in descriptive statistics is about pointing out the input variables, which most strongly affect the response. For a given data set of the type S_{O_1} it indicates which of the TF/IDF vector dimensions are most important for the separation of the documents into such that belong to a given concept and such that do not. Our variable selection criterion is the sensitivity of the VC dimension [3] - an indicator of the classifying capacity of a set of classifiers (e.g. the set of hyperplanes in a multidimensional space). Our initial experiments have shown variations in the estimation of that parameter according to the presence or absence of a given variable (vector dimension) in the data set. The procedure yields for each ontology an ordered list of variables on top of which are found the variables which are most important for the class separation. If the orders of the variables in both sets are similar, or if a significant number of most pertinent variables from both sets coincide, then the two concepts A and B are identified as similar.

3 A Procedure for Ontology Mapping

The structural and extensional approaches described so far are our instruments used to build a combined procedure for ontology mapping. Another important criterion for concept similarity is the presence of similar concept names in both ontologies. Linguistic analysis approaches to this problem, relying on names and textual description of ontology elements, are used in [2] and [9]. Even though not explicitly discussed in this section, we keep in mind that this name-based criterion is to be checked at any step before measuring the instance-based similarity of a pair of concepts and is to become an integral part of the structural similarity approach.

Let us take as input again the ontologies O_1 and O_2 together with their corresponding document sets D_{O_1} and D_{O_2}. In the following, we describe our method for combining the mapping approaches earlier.

Case 1: $|D_{O_1}| \approx |D_{O_2}|$

The first big case considers ontologies which contain similar number of documents. The ratio $r_\Delta = \frac{|D_{O_1} \cap D_{O_2}|}{|D_{O_1} \cup D_{O_2}|}$ is an indicator of the size of the intersection of both sets relative to the sets size. There are two further possibilities:

- **Case 1.1.** $r_\Delta > c_{r_\Delta}$, where $c_{r_\Delta} \in (0, 1)$ is a parameter to be fixed. In this case we have two different ontologies on (almost) the same documents sets. It is very likely that they share a conceptual similarity. We proceed to checking the graph distance between them.

- *Case 1.1.a.* $d(G_{O_1}, G_{O_2}) \approx 0$. The taxonomies have similar structures, describe the same domain and have the same extensions. It is left to establish the precise concept-to-concept mappings, done by the help of the *instance based* similarity check.

- *Case 1.1.b.* $d(G_{O_1}, G_{O_2}) \approx 1$. The maximal common subgraph of both ontologies is quite small, i.e. one of the taxonomies contains significantly lower number of nodes compared to the other (Figure 2(a)). Let us assume that $|C_{O_1}| < |C_{O_2}|$. Since both ontologies are "built" on approximately the same sets of documents, this means that O_2 is more specific than O_1, and contains more concept nodes. O_1 can be directly injected into O_2. The concept-to-concept correspondences indicating the exact injection pattern are provided by instance-based concept similarity applied on the set of the nodes of the *mcs* of both taxonomies.

- **Case 1.2.** $r_\Delta \leq c_{r_\Delta}$. The ontologies are little likely to be similar since their extensions share very little (or no) overlap.

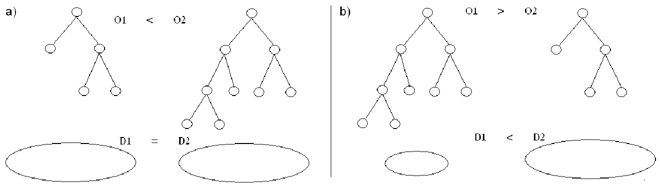


Fig. 2. a) Case 1.1.b. b) Case 2.3.b.2)

Case 2: $|D_{O_1}| < |D_{O_2}|$

In the second big case, the set D_{O_1} contains less documents than the set D_{O_2} (conventional choice). One can distinguish between three further sub-cases: Case

2.1. - the two sets do not intersect ($r_\Delta = 0$); Case 2.2. - the two sets intersect, but do not fully overlap; and Case 2.3. - the smaller set is a subset of the bigger one. Case 2.1. is in conflict with a major assumption introduced in the beginning and therefore does not provide mapping candidates. Case 2.2. conforms with either Case 2.1. or Case 2.3., depending on the size of the intersection $D_{O_1} \cap D_{O_2}$ relative to $|D_{O_1}|$. We will study in details Case 2.3. and proceed to measure the structural similarity between the inputs.

- *Case 2.3.a.* $d(G_{O_1}, G_{O_2}) \approx 0$. O_1 is structurally very similar to O_2. Hence, it is just as specific as O_2, but less populated with documents. This indicates that O_1 can be replaced entirely by O_2.

- *Case 2.3.b.* $d(G_{O_1}, G_{O_2}) \approx 1$. There are two different scenarios, depending on which of the two input ontologies contains more nodes.

1) $|C_{O_1}| < |C_{O_2}|$, i.e. there are less concepts in O_1 than in O_2. This is the case when O_1 is a sub-taxonomy of O_2 and can be entirely injected into it, as described in Case 1.1.

2) $|C_{O_2}| < |C_{O_1}|$. O_1 is more granular a hierarchy, but less populated than O_2 (Figure 2(b)). We will take instances from O_2 and assign them to O_1 by first aligning the nodes of both ontologies by the help of the *instance-based* mapping procedure. to another in terms of both conceptualization and instantiation.

References

1. H. Bunke, K. Shearer. A graph distance metric based on the maximal common subgraph, *Pattern Recogn. Lett.*, volume 19, number 3-4, 255–259, 1998.
2. P. Cimiano, A. Hotho, S. Staab. Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis, *JAIR Volume 24*, 305-339, 2005.
3. N. Cristianini, J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.*, Cambridge University Press, ISBN 0-521-78019-5, 2000.
4. A. Doan, J. Madhavan, P. Domingos, A. Halevy. Learning to map between ontologies on the semantic web, *WWW '02: Proceedings of the 11th international conference on World Wide Web*, 662–673, 2002.
5. J. Euzenat, P. Shvaiko. *Ontology Matching*, Springer-Verlag New York, Inc., 2007.
6. A. Isaac, L. van der Meij, S. Schlobach, S. Wang. An empirical study of instance-based ontology matching. In *Proceedings of the 6th International Semantic Web Conference*, Busan, Korea, 2007.
7. T. Joachims. Text categorization with support vector machines: learning with many relevant features. *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Number 1398, 137-142, 1998.
8. R. Korfhage. *Information Storage and Retrieval*, Section 5.7, Document Similarity, 125-133. Wiley and Sons, 1997.
9. G. Stumme, A. Maedche. FCA-MERGE: Bottom-Up Merging of Ontologies, *IJCAI*, 225-234, 2001.
10. K. Todorov. Combining Structural and Instance-Based Ontology Similarities for Mapping Web Directories, *ICIW, Third International Conference on Internet and Web Applications and Services, Athens*, 596-601, IEEE, 2008.
11. G. Valiente. *Algorithms on Trees and Graphs*, Springer-Verlag, 2002.

Testing the Impact of Pattern-Based Ontology Refactoring on Ontology Matching Results

Ondřej Šváb-Zamazal[1], Vojtěch Svátek[1], Christian Meilicke[2], and Heiner Stuckenschmidt[2]

[1] University of Economics, Prague, Dept. Information and Knowledge Engineering
{ondrej.zamazal,svatek}@vse.cz
[2] University of Mannheim, KR & KM Research Group
{christian,heiner}@informatik.uni-mannheim.de

Abstract. We observe the impact of ontology refactoring, based on detection of name patterns in the ontology structure, on the results of ontology matching. Results of our experiment are evaluated using novel logic-based measures accompanied by an analysis of typical effects. Although the pattern detection method only covers a fraction of ontological errors, there seems to be a measurable effect on the quality of the resulting matching.

1 Introduction

Ontologies in formal languages often suffer from diverse kinds of modeling errors in the structure and/or naming style. These errors can typically be perceived as violation of the set-theoretic interpretation of the subclass relationship. We hypothesize that if we repair some of those errors in OWL ontologies as a pre-processing step for OM, we will get better results from OM tools than with original unrepaired OWL ontologies.

This hypothesis has been evaluated by an experimental study. The whole experiment is depicted in Figure 1. Having detected modeling errors via name structure analysis, we apply several *refactoring operations* on them. Although 'ideal' refactoring can in principle be arbitrarily complex, we found three generic refactoring operations that seem to cover a significant number of realistic cases [4]. The result of mapping a pair of ontologies that underwent refactoring is compared with the result of mapping the same pair of ontologies in the original form.

Section 2 deals with patterns detected in ontologies and refactoring operations that are described with several examples. Some statistics about frequencies of patterns and refactoring in our experiment as well as setting of the experiment, an evaluation method, and results of experiment are described in section 3. The paper is wrapped up with conclusions and future work.

2 Patterns and Refactoring Operations

The patterns for the study were chosen based on our preliminary manual analysis of numerous ontologies, and thus correspond to generalisations of 'striking' fragments of real ontologies (the inventory of patterns is thus definitely not complete and will be

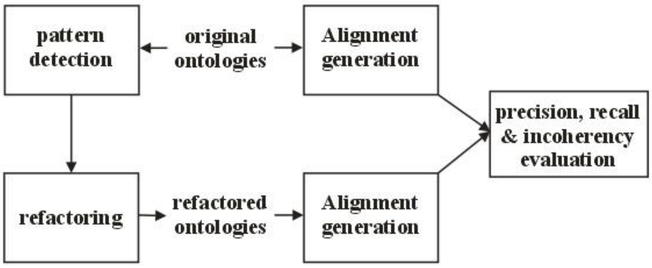


Fig. 1. Workflow of experiment

extended by future research). Our approach to pattern detection has essentially been built upon the notion of *named structural cluster*. Theoretically, an ideal OWL ontology could consist of large named structural clusters going from the upmost levels of the hierarchy to the leaves, as the type of entity should not change when going down the tree—it can only be refined, which is often done by extending the original name. In reality, however, such large named structural clusters are rare. Clusters can legally be broken by introducing lexical hyponymy (and possibly synonymy) into head noun naming; using thesauri could help here to some degree. Inadequate breaks however often appear due to either bad naming practices or to inherent errors in conceptualization.

For the sake of brevity we concisely describe and exemplify our three patterns, which have been described using formal framework in [4]. Furthermore, we illustrate three basic refactoring operations as they have been used in our experiment.

First pattern *SE* (matching siblings with non-matching parent) represents the situation that two or more children do not have the same head noun as their parent but have the same head noun among themselves. This pattern might indicate an *overly flat* hierarchy, asking for inclusion of an intermediate concept superordinated to some of the sibling classes only. It can also be produced by a modeling error or by awkward naming. In this case we can employ a *renaming operation* (RN), which renames the children in a suitable way, eg. appending a head noun of parent after the presumed head noun of children (see Figure 2). Other option would be a substitution the presumed head noun of each of children with the presumed head noun of parent.

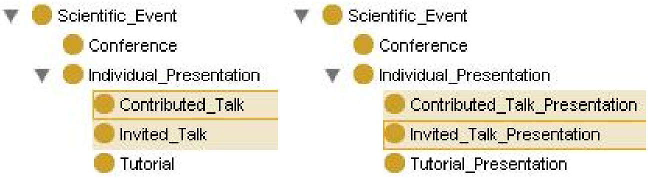


Fig. 2. Example of RN (ekaw.owl)

Pattern *hE* (plain head noun) represents the situation that two or more children do not have the same head noun as their parent but have the same head noun among themselves and one of them is plain head noun, i.e. there is no other word in name but the head noun. Actually, this is the specific case of SE pattern. In this case we can employ a *restructuring operation* (RS) that leads to shift a concept that is badly placed in the taxonomy, eg. a concept with plain head noun should be subclass of parent with the same head noun.

Finally, pattern *ME* (matching outlier) represents the situation that a concept shares the head noun with a cluster that it is not descendant of (it is a subclass of any of the concepts from the cluster). In this case we can use an *operation of adding a concept* (ADD) that leads to adding a new concept into the taxonomy, eg. in Figure 3 for reconciliation of the ontology we employ two operations: first ADD and then RS.

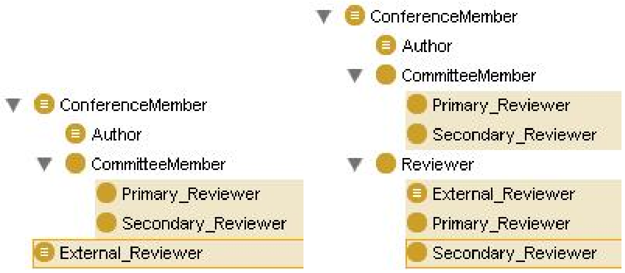


Fig. 3. Example of ADD (iasted.owl)

All cases of modelling errors detected via some of above mentioned patterns can be repaired by one of three refactoring operations: RN, RS, or ADD. It depends on a situation which one is the most suitable. It is also possible to employ more than one operation for one case.

3 Experimental Evaluation

For our experiment we chose seven ontologies from the OntoFarm collection[3] describing the domain of conference organization. We manually refactored these ontologies according to the name patterns discussed above, which are detected automatically. In Table 1 we can see how often these patterns have been detected as well as the number of refactoring operations applied to each pattern.

We automatically generated alignments for five pairs of ontologies, namely the ontology pairs *cmt-ekaw*, *confOf-sigkdd*, *ekaw-iasted*, *ekaw-sigkdd*, and *myReview-edas*. For each matching problem we applied three matching tools for both the original ontologies and their refactored counterparts. We have chosen *Falcon-AO* [2], *HMatch* [1],

[3] http://nb.vse.cz/~svatek/ontofarm.html

Table 1. Frequencies of patterns and refactoring operations.

	Ontologies							Refactoring		
	cmt	ekaw	confOf	sigkdd	iasted	myReview	edas	RN	RS	ADD
SE	1	2	1	5	8	1	3	10	2	9
hE	-	-	1	-	-	1	-	-	2	-
ME	1	2	-	1	1	-	-	2	1	2

and *ASMOV* [5] as representative matching systems. Since our refactoring approach is currently limited to concepts, we only considered correspondences between concepts. In addition to classical evaluation methods and a discussion of some examples we also applied the maximum cardinality incoherence measure defined in [3].

For Falcon-AO the effects of refactoring are very similar across all ontology pairs. Falcon-AO generates between 5 and 12 correspondences with respect to the original ontologies and most of these correspondences are correct. Thus, it is no suprise that these alignments were coherent before and after the refactoring with one exception. Due to the refactoring, for each matching pair one more correspondence has been generated. Notice that the increased size of the alignments had no negative impact on their coherence. In particular, all additionally found correspondences have been verified as correct. In most cases we observed that these effects are based on the refactoring strategy of introducing an additional concept into the conceptual hierarchy to repair the SE pattern. Since there often exists a counterpart to the additionaly introduced concept, a new correspondence can now be detected.

Since HMatch generates less coherent alignments, it made sense to compute the average of the incoherence degree. Comparing the mappings created for the original and the refactored ontologies, we could observe a decrease of the incoherence by 24%[4]. This difference can partially be explained by a closer look at one of the alignments. Matching ontology *myReview* with *edas* generates amongst others correspondence $\langle \mathit{myReview\#Chair}, \mathit{edas\#ConferenceChair}, =, \mathit{0.56} \rangle$. Due to the refactoring of pattern SE we introduced concept $\mathit{edas\#Chair}$ as parent of $\mathit{edas\#ConferenceChair}$. HMatch now finds a better matching counterpart for $\mathit{myReview\#Chair}$ and replaces the incorrect correspondence by a correct correspondence. This is a typical example where both precision and recall are increased by a refactoring operation.

The results for ASMOV are less clear-cut. In particular, we found that a significant part of the alignment changed due to the refactoring (compared to the other systems). Although we were not able to detect a continuous pattern, we observed that the refactoring had the strongest positive effect on matching *myReview* with *edas* where the degree of incoherence was reduced by 47%[5]. A closer look revealed an interesting pattern. The alignment based on the original ontologies contains amongst others correspondences (1) $\langle \mathit{myReview\#Document}, \mathit{edas\#Document}, =, \mathit{0.68} \rangle$ and (2) $\langle \mathit{myReview\#CD\_ROM}, \mathit{edas\#ReviewForm}, =, \mathit{0.52} \rangle$. Contrary to this, the alignment generated based on refactoring did not contain the incorrect correspondence (2). This is a surprise at first sight, because none of the refactoring operations was directly concerned with $\mathit{myReview\#CD\_ROM}$. Actually, we applied a restructuring opera-

[4] From 0.108 to 0.082

[5] from 0.105 to 0.056

tion by adding the axiom $myReview\#OutputDocument \sqsubseteq myReview\#Document$. Together with the disjointness axiom $edas\#ReviewForm \sqsubseteq \neg edas\#Document$ that is given in the *edas* ontology, the ASMOV system detects a conflict between correspondence (1) and (2) in its validation phase. Due to the semantics induced by the restructuring operation, it can be detected that (1) and (2) are mutually exclusive.

Overall, we conclude that refactoring improves the quality of an alignment generated by a matching system. Five of the generated alignments have been incoherent before the refactoring has been applied. For four of these alignments we measured a decrease of incoherence, while none of the coherent alignments becomes incoherent. More important is the result that refactoring increases both precision and recall in many cases. In particular, the last example showed that it is possible to use the additional information induced by the refactoring in a non trivial way to filter out incorrect correspondence.

4 Conclusions and Future Work

In our work we attempted to combine two seemingly distant areas: ontology mapping evaluation and ontology refactoring. We hypothesized that OM tools will reach better results for repaired OWL ontologies than for original unrepaired OWL ontologies. This hypothesis was to some degree confirmed by our experiment. We carried out the experiment over a complex workflow, starting with automatic detection of patterns potentially indicating conceptualisation errors through manual refactoring and application of several off-the-shelf matching tools up to mapping evaluation accompanied by a detailed analysis of the most interesting examples. In future work, the set of detectable patterns and refactoring operations will be adjusted and extended. In particular, we have to make our approach applicable to properties. A consolidated description framework for patterns and refactoring might also allow to partially automate the refactoring. In an automated setting at least the restructuring operation requires to be validated by logical reasoning to avoid the introduction of logical inconsistencies.

Acknowledgments The research was partially supported by the IGA VSE grant no.20/08 “Evaluation and matching ontologies via patterns” and by the German Science Foundation (DFG) in the Emmy Noether Programme under contract STU 266/3-1

References

1. Castano S., Ferrara A., Montanelli S.: Matching Ontologies in Open Networked Systems: Techniques and Applications. *Journal on Data Semantics*, 2006.
2. Hu W., Qu Y.: Falcon-AO: A Practical Ontology Matching System. *Journal of Web Semantics*, 2007.
3. Meilicke C., Stuckenschmidt H.: Incoherence as a Basis for Measuring the Quality of Ontology Mappings. In: OM Workshop 2008.
4. Šváb-Zamazal, O. and Svátek, V.: Analysing Ontological Structures through Name Pattern Tracking. In: EKAW 2008, Acitrezza, Italy, Springer LNCS.
5. Jean-Mary, Y. R., Kabuka, M. R.: ASMOV results for OAEI 2007. In: OM Workshop 2007.

Relevance-Based Evaluation of Alignment Approaches: the OAEI 2007 food task revisited

Willem Robert van Hage[1,2], Hap Kolb[1], and Guus Schreiber[2]

[1] TNO Science & Industry, Stieltjesweg 1, 2628CK Delft, the Netherlands,
hap.kolb@tno.nl
[2] Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1081HV Amsterdam, the Netherlands,
wrvhage@few.vu.nl, schreiber@cs.vu.nl

Abstract. Current state-of-the-art ontology-alignment evaluation methods are based on the assumption that alignment relations come in two flavors: correct and incorrect. Some alignment systems find more correct mappings than others and hence, by this assumption, they perform better. In practical applications however, it does not only matter *how many* correct mappings you find, but also *which* correct mappings you find. This means that, apart from correctness, relevance should also be included in the evaluation procedure. In this paper we expand the sample-based evaluation of the OAEI 2007 *food task* with a sample evaluation that uses relevance to prototypical search tasks as a selection criterion for the drawing of sample mappings.

1 Introduction

In recent years ontology alignment has become a major field of research [3, 5]. Especially in the field of digital libraries it has had a great impact. Good evaluation is essential for the deployment of ontology-alignment techniques in practice. The main contribution of this paper is to offer a simple method to capture the performance of alignment approaches in actual applications. We introduce *relevance-based evaluation*, which compensates for some of the shortcomings of existing methods by using the needs of users during sample selection. We apply this method to the data of the OAEI 2007 *food task* [2].

Nearly all existing evaluation measures used to determine the quality of alignment approaches are based on counting mappings [1, 2]. For instance, in the context of ontology alignment, the definition of Recall is defined as the number of correct mappings a system produces divided by the total number of correct mappings that can possibly be found (*i.e.* that are desired to be part of the result). Regardless of their differences, most of these measures have one thing in common: They do not favor one mapping over the other in order to give an objective impression of system performance. Any mapping could prove to be important to some application. Therefore, they can only tell us *how many* mappings are found on average by a system, but not *which* mappings are found and whether the mappings that are found are those that are useful for a certain application. Whenever someone wants to decide which alignment approach is best suited for his application (*e.g.* [7]) he will have to reinterpret average expected performance in the light of his own needs. This can be a serious obstacle for users.

A solution to this problem is to incorporate the importance of mappings (*i.e.* relevance) into the evaluation result. This solution immediately raises two new problems: (1) How to come up with suitable importance weights, and (2) How to define a simple and intuitive way to use these weights With respect to problem 1, there are many sensible ways to weigh the importance of mappings. For example, based on the size of the logical consequence, *cf.* Semantic Precision and Semantic Recall [1], or on expected traversal frequency, *cf.* [4]. Relevance-based evaluation equates importance to relevance to prototypical application scenarios. Likewise, with respect to problem 2, there are many sensible ways to incorporate mapping importance into an evaluation method. For example, linear combination, *cf.* [6], or stratification *cf.* [9]. As opposed to existing methods to account for the relevance of mappings that include it as a variable in an evaluation measure, we use relevance to steer the sample-selection process. Instead of randomly selecting mappings for the evaluation of alignment approaches (*cf.* the *food* and *environment tasks* described in [2]) we select *only* those that are relevant to an application. This way we can use existing and well-understood evaluation metrics, like Precision and Recall, to measure performance on important tasks as opposed to fictive average-case performance.

2 Experimental Set-up

We demonstrate how relevance-based evaluation works by extending the existing results of the OAEI 2007 *food task*, which did not take relevance into account. We determine relevance for the mappings based on hot topics related to this task, like global warming and increasing food prices, which we obtain by means of query-log analysis, expert interviews, and news feeds. For the original OAEI 2007 *food task*, Recall was measured on samples that represent the frequency of topics in the vocabularies. In relevance-based evaluation the samples are drawn by the frequency of use in search tasks, specifically, finding documents about prototypical agricultural topics of current interest in one collection using the indexing vocabulary of the other. The procedure we use is as follows: (1) Gather topics that represent important use cases. We gather "hot" topics in agriculture from the query log files of the FAO AGRIS/CARIS search engine, the FAO newsroom website, and interviews with two experts. Patricia Merrikin from the FAO's David Lubin library, and Fred van de Brug, from the TNO Quality of Life food-safety group. We manually construct search-engine queries for each topic. (2) Gather documents that are highly relevant to these topics. We ascertain which documents would be sufficient for the hot topics by gathering suitable candidate documents from the part of the FAO AGRIS/CARIS and USDA AGRICOLA reference databases that overlaps. We use a free-text search engine[3] and manually filter out all irrelevant documents. (3) Collect the meta-data describing the subject of these documents and align the concepts that describe the subject of the documents to concepts in the other thesaurus. We collect values of the Dublin Core subject field from the AGRIS/CARIS and AGRICOLA reference databases. These values come from subject vocabularies, respectively AGROVOC and the NAL Agricultural Thesaurus. We manually align each concept to the most similar concept in the other vocabulary. The resulting mappings make up our sample set

[3] `http://www.fao.org/agris/search`

of relevant mappings. (4) Apply the mappings for evaluation by counting how many of these mappings have been found by ontology alignment systems and comparing system performance based on these counts. Specifically, we re-calculate Recall for the top-4 systems of the OAEI 2007 *food task*, following the same procedure as described in [2, 9], but use the new set of relevant mappings.

3 Sample Construction

Topics In order to get a broad overview of current affairs in the agricultural domain we gathered topics from three sources: Analysis of the search log files of the AGRIS/CARIS search engine, topics in the "Focus on the issues" section of the FAO Newsroom, and expert interviews. Detailed descriptions of the topics can be found at `http://www.few.vu.nl/~wrvhage/om2008/topics.html`.

Documents Per topic did a full-text search on the AGRIS/CARIS search engine limited to the set of documents that is shared between the AGRICOLA and AGRIS/CARIS collections and fetched the top-100 of the results. From these 1500 documents we selected only the ones that are relevant to our topics, on average 31 per query, and that have been assigned Dublin Core subject terms in both collections. This left 52 documents in total, on average 3.8 per query. For four of the topics we found no documents that were both relevant and indexed in both collections. The reason for this is that these topics are all very new issues. The greatest overlap between the AGRIS/CARIS and AGRICOLA collections exists for documents published between 1985 and 1995. After the year 2000 no documents have been imported and thus it is hard to find relevant documents for new issues. We assume that the 52 double-annotated relevant documents are representative of the set of all relevant documents with subject meta-data, *i.e.* also the documents with only annotations in one of the two collections. These are the documents for which alignment could make the biggest difference. This is a reasonable assumption, because the indexing process of both collections is regulated by a protocol to control continuity.

Mappings Having established which documents are potentially important to find, we have to decide which mappings will be of most benefit to someone who wants to find them. We assume that the mappings that map the subject annotations as strictly as possible to the other vocabulary are the most beneficial for any search strategy that employs them. Given this assumption, we manually constructed the set of mappings that connect each concept used to index the 53 relevant documents with its most similar counterpart.

The alignment of the 266 NALT concepts and 212 AGROVOC concepts was done by thesaurus experts at the FAO and USDA, Gudrun Johannsen and Lori Finch. This led to a sample reference alignment consisting of 347 mappings: 74 broadMatch / narrowMatch and 273 exactMatch (79%). 11 concepts had no exact, broader or narrower counterpart. This is a higher percentage of exactMatch mappings than we expected based on our experiences with the OAEI *food task*. For the *food task*, arbitrary subhierarchies of the AGROVOC and NAL thesaurus were drawn and manually aligned with the other thesaurus. Most of the resulting mappings were equivalence relations. The sample sets, the percentage of equivalence mappings in the reference alignment (*i.e.* the desired equivalence relations) varied between 54% and 71%.

4 Sample Evaluation Results

Having constructed a new sample reference alignment we can use it to measure the performance of alignment approaches. The measurement of Recall under the open-world assumption is inherently hard, so we choose to reiterate the evaluation of Recall on the OAEI 2007 *food task*. This gives us a second opinion on the existing evaluation. For the sake of simplicity we calculate Recall scores of the top-4 of the systems that participated in the OAEI 2007 *food task*. The results are shown in table 1.

	Falcon-AO	RiMOM	DSSim	X-SOM
OAEI 2007 food, only exactMatch (54% of total)	0.90	0.77	0.37	0.11
hot topics, only exactMatch (79% of total)	0.96 ↑	0.60 ↓	0.16 ↓	0.07 ↓
OAEI 2007 food, exact, broad, narrowMatch	0.49	0.42	0.20	0.06
hot topics, exact, broad, narrowMatch	0.75 ↑	0.47 ↑	0.12 ↓	0.05 ≈

Table 1. Recall of alignment approaches measured on sample mappings biased towards relevance to hot topics in agriculture and on impartial, non-relevance-based sample mappings from the OAEI 2007 *food task*. Arrows indicate significant differences (using the tests described in [9]).

There are a number of striking points to note about these results. For most systems there is a significant positive or negative difference. Overall, the difference with non-relevance-based evaluation is large. For exactMatch relations performance in general is lower for relevance-based evaluation than for non-relevance-based evaluation, with the exception of Falcon-AO, although the relative difference is small. However, the ranking of the alignment approaches is left unchanged. The results of relevance-based evaluation seem to exaggerate the differences between the performance of the approaches. This can be explained by the relatively high number of obvious matches (93%) in the set of mappings on hot topics. None of the approaches was able to find a substantial number of difficult mappings, but the best approaches were good at finding all obvious mappings before resorting to speculation about the harder mappings. The best two systems, Falcon-AO and RiMOM performed relatively good when accounting for all relation types, the last row of table 1, even though they found no broadMatch and narrowMatch relations. This is due to the kind of exactMatch relations they *did*, which were mostly of the obvious kind (*i.e.* literal matches), which was exactly the kind that was needed most for the hot topics. The high percentage of exactMatch relations in the set on hot topics accentuates their behavior. The converse goes for DSSim, which found a relatively low number of obvious mappings. Fewer broadMatch and narrowMatch mappings seem to be needed than one would expect from the non-relevance-based evaluation method. Compare the percentage in the OAEI 2007 Recall set, 54%, to the percentage based on hot topics, 78.6%. Although there is a large part of the AGROVOC and NALT vocabularies that does not have a counterpart in the other vocabulary, the portion that is actually used suffers less than one would expect from this mismatch. Apparently, indexers mainly pick their terms from a limited set, which shows a greater overlap. (After all, why needlessly complicate things?) It remains to be seen if this also applies to other vocabulary mappings. On one hand this means that approaches that can only find equivalence mappings perform better in practice than was expected. On the other hand it confirms the expectation that a large part (*more than 20%*) of the mappings that are needed for federated search over AGRIS/CARIS and AGRICOLA consists of other relations than equivalence relations. Also, one can conclude that systems that are incapable of finding a substantial number of equivalence relations can only play a marginal role.

5 Discussion

By using relevance as a sample criterion we avoid having to come up with an artificial approximation of importance. We can simply explore the performance difference on samples consisting of relevant mappings and samples consisting of irrelevant mappings. Under minimal assumptions we avoid having to choose a specific retrieval method while retaining the the character of an end-to-end evaluation. (*cf.* the *End-to-end Evaluation* method described in [9]) This saves us the effort of extensive user studies while not ignoring the behavior of alignment approaches in real-life situations. Considering the fact that AGROVOC and NALT are two of the most widely used agricultural ontologies, and that they are prototypical examples of domain thesauri in their design we conclude the following. From the point of view of a developer of a federated search engine in the agricultural domain that needs an alignment we can conclude that at the moment the Falcon-AO is a good starting point. In the case described in this paper, Falcon-AO found three quarters of the mappings. This empirical study has shown that at least 20% of the required mappings to solve the typical federated-search problem described in this paper are hierarchical relations. Even though this is a smaller fraction than we initially expected it is still a large part. An extended version of this paper can be found in [8].

References

1. Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proc. of IJCAI 2007*, pages 348–353, 2007.
2. Jérôme Euzenat, Malgorzata Mochol, Pavel Shvaiko, Heiner Stuckenschmidt, Ondřej Šváb, Vojtěch Svátek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative, 2007.
3. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007. ISBN 978-3-540-49611-3.
4. Laura Hollink, Mark van Assem, Shenghui Wang, Antoine Isaac, and Guus Schreiber. Two variations on ontology alignment evaluation: Methodological issues. In *Proc. of ESWC*, 2008.
5. Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The knowledge engineering review*, 18(1):1–31, march 2003.
6. Jaana Kekäläinen. Binary and graded relevance in ir evaluations–comparison of the effects on ranking of ir systems. *Information Processing and Management*, 41(5):1019–1033, 2005.
7. Malgorzata Mochol, Anja Jentzsch, and Jérôme Euzenat. Applying an analytic method for matching approach selection. In *Proc. of OM-2006*, pages 37–48, 2006.
8. Willem Robert van Hage. *Evaluating Ontology-Alignment Techniques*. PhD thesis, Vrije Universiteit Amsterdam, 2008. http://www.few.vu.nl/∼wrvhage/thesis.pdf.
9. Willem Robert van Hage, Antoine Isaac, and Zharko Aleksovski. Sample evaluation of ontology-matching systems. In *Proc. of EON*, 2007.

Matching ontologies for emergency evacuation plans

Luca Mion[1], Ivan Pilati[1], David Macii[2], and Fabio Andreatta[3]

[1] TasLab – Informatica Trentina S.p.A.
{luca.mion,ivan.pilati}@infotn.it, http://www.taslab.eu
[2] Dept. of Information Engineering and Computer Science, University of Trento
macii@disi.unitn.it
[3] Dept. of Civil Protection, Autonomous Province of Trento
fabio.andreatta@provincia.tn.it

Abstract. In case of emergency, the coordination of different services deals with different working methods, different languages, different instruments, different sensors and different data representations. Thus, the coordination of services includes heterogeneity problems that can be managed with the help of ontology matching techniques. In this paper we present a scenario where the requirements for ontology matching arise from emergency evacuation plans, in the specific domain of civil protection applications. We envisage what kind of smart sensor technologies could be used to support critical decisions when heterogeneous sources of information have to be matched.

1 Introduction

In the context of semantic web and web services, heterogeneity represents a key feature. One of the critical issues of semantic web services is the way the resources of the semantic web have to be integrated as a whole. In fact, the ontologies that are used to express information by means of sets of discrete entities (e.g., classes, properties, rules) are affected by heterogeneity, which requires proper integration techniques [1, 2]. *Ontology matching*, namely the ability of finding suitable relationships between entities from different ontologies, is essential to achieve semantic interoperability and it may have huge social impact.

For example, when a large–scale disaster occurs many people from different organizations may reach the critical area in a short time, and the need for integration of heterogeneous and rapidly evolving sources of data emerges. In this context disaster management is strongly related, at different levels of abstraction, to environmental monitoring and ambient computing [3]. From a practical point of view, the monitoring of an area involved in a disaster can be regarded as a special example of environmental monitoring. In general, environmental monitoring applications require sensing different quantities (e.g., temperature, moisture or brightness), possibly evolving in time and space, as well as some information related to the physical context in which some services are available [4]. The purpose of these services can be merely informative, or aimed at making decisions. In situations where an impending danger may affect the life of several people at the same time, the criticalness of decisions and the dynamics of all

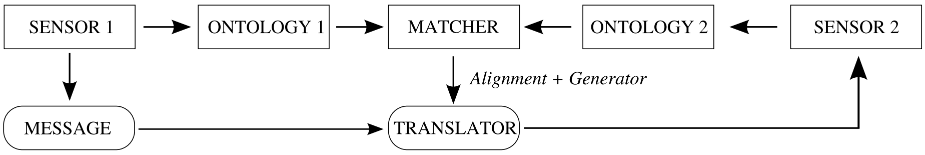


Fig. 1. Agent communication, adapted from [6].

events must be analyzed effectively in real-time, and thus the evolution in time and space of interesting quantities should be monitored as well.

In order to design flexible services in such extreme conditions, defining the ontologies of the various smart devices employed (along with their capabilities) and implementing suitable matching strategies is a viable and effective approach [6]. Heterogeneous ontologies have to get in correspondence in order to understand messages sent by related sensors. Sensors can perform ontology matching by themselves or by taking advantage of alignment or matching services, and when they find a mutual agreement they can transform the alignment in a program that translates the messages in axioms enabling the interpretation of the messages, as represented in Fig. 1.

This paper presents a potential real-world scenario where the ontology matching requirements are related to the management of emergency evacuation plans from large buildings. In particular, we envisage how management of emergency evacuation plans from large buildings can take advantage from ontology matching. The case study considered in the following, although still at a quite preliminary stage, results from a close collaboration between the Civil Protection Department, various research centers and local companies, and with the help of both staff members and volunteers of various rescue corps.

The rest of the paper is organized as follows: Section 2 introduces the basic issues related to emergency situations and specifically describes the requirements of the emergency evacuation plan scenario posed to ontology matching. Then, Section 3 summarizes the conclusions and outlines future work.

2 Matching ontologies in emergency situations

In case of emergency, an effective coordination of people from different organizations (e.g., civil protection, police, ambulance, fire brigades, red cross) is essential. The services offered by such organizations are characterized by different working methods, different languages, different instruments, sensors, and data representations. Thus, the coordination of services certainly includes problems requiring ontology matching. Several monitoring systems can indeed collect data for different organisations and for different purposes, using different sensor technologies.

Although the different solutions deal with huge amount of data, the interpretation and analysis is not consistent. Proper standardisation of data collection processes is necessary, including applied technology and data storing formats, to facilitate communication between services on a more consistent basis. Also, un-

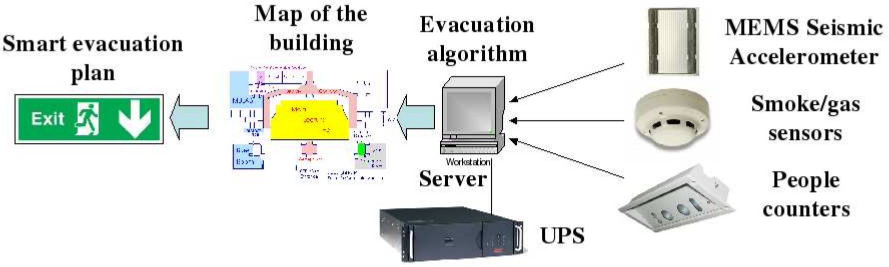


Fig. 2. Example of sensor communication and ontology matching based on sensor ontologies in order to facilitate decision making.

certainty over the network consistence arises during emergencies (e.g., in terms of sensors full functionality, or possible interaction that may occur with robots or automated agents with own sensors), as exemplified in Fig. 2.

2.1 Emergency evacuation plans

As presented in Sec. 1, applications evolve in changing environments where devices are replaced and added, and then it is not possible to establish unique and definitive ontologies. Thus, applications have to be expressed in terms of generic features that are matched against the actual environment. An interesting civil protection applicative scenario, in which smart sensor networks could be particularly useful, concerns with the optimal management of emergency evacuation plans from large buildings. As known, all public buildings (e.g., offices, shopping malls, schools) are usually equipped with a certain number of safety exits as well as by evacuation plans that should be carefully respected, in particular in case of dangerous events (e.g., fires or earthquakes). Usually, the basic safety requirements are defined by national regulations. However, the actual effectiveness of any pre-set evacuation plan can be limited by several issues, such as the impossibility for occasional visitors (e.g., customers) to know the evacuation strategy, the unpredictability of crush behavior in panic conditions, the lack of information about number and about the distribution of people inside the building. To solve such problems a smart sensor network could be deployed in a building in order to automate the whole evacuation process. This network might consist, for instance of:

- Redundant smoke or gas sensors to detect the presence of fire or the risk of an explosion (redundancy is essential in this case to reduce the risk of false alarms);
- Low-cost micro-electro-mechanical accelerometers for seismic events monitoring [8] placed along the most important architectural elements of the building;
- Smart video people counters located in proximity to doors, stairs or corridors in order to estimate in real–time not only the total amount of people in the building, but also their distribution [9–11].

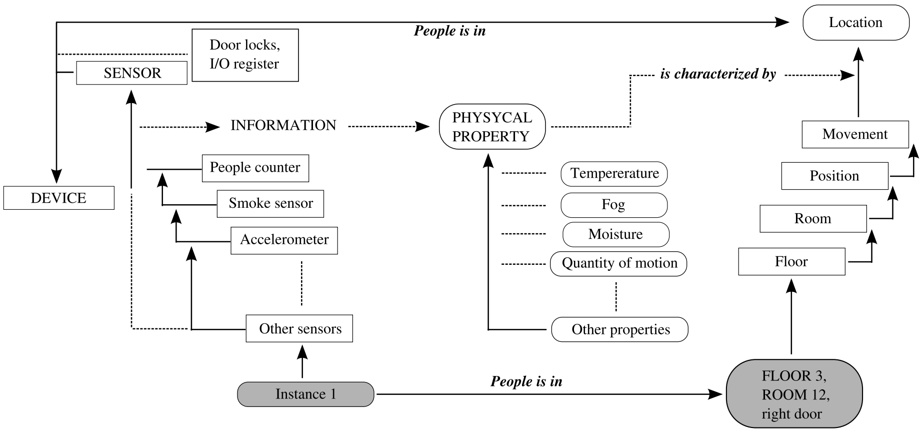


Fig. 3. Ontologies related to the objects found in the building during evacuation, inspired by [12], with instances linked to their classes.

The data collected by the various sensors could be transferred through wired or wireless connections (e.g., Ethernet or Wifi) to a central server (suitably connected to an emergency Uninterruptible Power Supply, UPS) which in turn could activate visual or sound alarms in order to manage the evacuation process in the safest possible way. For instance, when a fire is detected, a software application running on the server could estimate the level of risk in each area of the building and then switching on the emergency signals and the way-out light indicators, keeping into account the position and the distance of different users from the impending danger. In this way, people could be safely and orderly guided towards the safest exits, and in addiction the intrinsic risks related to a mass evacuation (especially for children, for elderly people and for people with disabilities) could be significantly reduced.

2.2 Emerging requirements

Usability of devices depicted in Fig. 2 is unpredictable since they are subjected to being added/replaced or malfunctioning at any time. For instance, when a robot enters a building during evacuation, it will introduce sensors that will provide more precision or information which has not been considered at application design time and again ontology description languages can help solving this problem [12]. Fig. 3 shows possible specific ontologies related to the objects found in the building during evacuation.

From this scenario we can derive requirements for matching solutions in the context of emergency evacuation plans. In particular, requirements concern specific behaviours, such as requirements of being automatic (not relying on user feedback), being correct (not delivering incorrect matches), being complete (delivering all matches) and being performed at run time. These requirements confirm the application requirements reported in [6], with reference to multi-agent communication. Another important requirement concerns the execution time, which has been indicated to be under 2 seconds by the Civil Protection staff, in order to operate under stable and safe conditions.

3 Conclusions

In this paper, an applicative example is proposed in which the joint application of both ontology matching strategies and smart sensor networks can be successfully used to optimize building evacuation. Multiple sensors could be used to estimate in real–time the total amount of people and their distribution in the building, while proper matching of the sensor ontologies should facilitate and greatly improve the decision making process. Future works include studies to elaborate and to formalize the scenario, to choose and to develop a suitable matching algorithm (e.g., as in [13]), and and extensive end-to-end testing.

Acknowledgments. We acknowledge the Autonomous Province of Trento for supporting TasLab Living Lab and the European Network of Living Labs [14] for promoting innovation activities in ICT.

References

1. F. Giunchiglia, M. Yatskevich, P. Avesani, and P. Shvaiko, "A Large Scale Dataset for the Evaluation of Ontology Matching Systems," Knowledge Engineering Review Journal, 2008.
2. J. Euzenat, "Alignment infrastructure for ontology mediation and other applications," in Proc. 1st ICSOC Int. Workshop on Mediation in semantic web services, pp. 81–95, 2005.
3. J. Carlos , A. Jun, and L. L. Chen, "Using Ambient Intelligence for Disaster Management," in: Knowledge-Based Intelligent Information and Engineering Systems (KES2006), pp. 171–178, 2006.
4. O. Khriyenko and V. Terziyan, "Context description framework for the semantic web," In Proc. Context 2005 Context representation and reasoning workshop, 2005.
5. TasLab Webpage. `http://www.taslab.eu`.
6. J. Euzenat and P. Shvaiko, "Ontology matching," Springer, Heidelberg (DE), 2007.
7. M. Marchese, L. Vaccari, P. Shvaiko, and J. Pane, "An Application of Approximate Ontology Matching in eResponse," In Proc. of ISCRAM, 2008.
8. REF TEK Webpage, `http://www.reftek.com`.
9. A. Bevilacqua, L. Di Stefano, and P. Azzari, "People Tracking Using a Time-of-Flight Depth Sensor," IEEE Computer Society, 2006.
10. Neuricam Webpage. `http://www.neuricam.com`.
11. F. Leonardi and D. Macii, "An Uncertainty Model for People Counters based on Video Sensors," Proc. IEEE International Workshop on Advanced Methods for Uncertainty Estimation in Measurement (AMUEM), pp. 62–66, 2008.
12. J. Euzenat, J. Pierson, and F. Ramparani, "Dynamic context management for pervasive applications," Knowledge engineering review, 23(1):21–49, 2008.
13. F. Giunchiglia, F. McNeill, M. Yatskevich, J. Pane, P. Besana, and P. Shvaiko, "Approximate structure-preserving semantic matching," In Proc. of ODBASE, 2008.
14. European Network of Living Labs Webpage. `http://www.openlivinglabs.eu`.

Ontological Mappings of Product Catalogues

Domenico Beneventano[1] and Daniele Montanari[2]

[1]Università di Modena e Reggio Emilia, Dipartimento di Ingegneria dell'Informazione
Via Vignolese 905, 41100 Modena, Italy
domenico.beneventano@unimore.it

[2]Eni S.p.A., ICT / Semantic Technologies Via Arcoveggio 74/2, 40129 Bologna, Italy
daniele.montanari@eni.it

Abstract. In this paper we built on top of recent effort in the areas of semantics and interoperability to establish the basis for a comprehensive and sustainable approach to the development and later management of bridging systems among a variety of corporate system that need to be interconnected without being individually modified. In particular, we collect some preliminary evidence that a sustainable approach exists to the definition of mappings which can withstand changes of the underlying classification schemes. This in turn adds evidence towards the feasibility of a dynamic interoperable infrastructure supporting a global adaptive electronic market place.

1 Introduction

The goal of this paper is to combine some results on the development of ontologies for products and services classification with other results in the area of system interoperability and ontology mapping to study the impact of evolution of the reference ontologies onto the catalogues/classification system annotated and then derivatively mapped w.r.t the ontologies. Slightly more formally, given comparable catalogues C1, C2, and assuming that OntologyA and OntologyB are reference ontologies which have been used to annotate the content of C1 and C2 respectively, given a mapping between OntologyA and OntologyB which provides a correspondence between concepts and relations in the two ontologies, a derived mapping can be defined at the level of the catalogues C1 and C2 (see Figure 1).

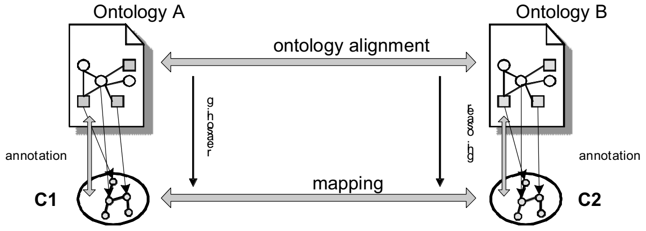


Figure 1 Mappings at reference and catalog levels [1]

It has been observed [7] that product and service ontologies exhibit a significant evolution of their content in time, due to changing market condition, and the evolving user sophistication and needs. This implies that the definition of the mapping between the catalogues will not be a one-time operation, but rather a repeated operation following the version cycles of the involved ontologies.

Being a heavy semi-automatic operation, the cost of the change of mapping must be carefully assessed, and understanding whether there are ways of minimizing the impact of these reviews can be a valuable information for people planning to position themselves, their products, and their services in an electronic market where they need to interoperate with other heterogeneous actors / systems.

This work was partially funded under the European Commission 6th Framework Programme under contract FP6-2005-IST-5-034980 (STASIS).

2 Ontologizing Product Catalogues

The first step when entering the semantic dimension of a field consists in providing a semantic reference for all the relevant entities in the domain. Several product and service classification schemes are available nowadays, both as in-house developments fulfilling the needs of their original users, and as more or less open public standards; some well known such schemes are UNSPSC [13], eCl@ss [6] and RosettaNet [11]. A good account of the subtleties lurking in the conversion from classification system into ontology is presented in [7]. One crucial point made is that the typical hierarchies of classification entities found in a classification system, being driven among other things, by the typical needs of purchasing departments in terms of searching, reporting, and classifying suppliers of goods and services.

Once we have reference ontologies derived from the standard classification systems, we can use them to annotate a given catalog of products and / or services. In [1] we describe a technique which derives an ontology for a specific database schema or semi-structured set of information (like web or XML pages); this technique was experimented in the STASIS project (http://www.stasis-project.net).

Let us introduce a real example of catalog by considering the eBay catalogue. This catalog is structured in three kinds of elements, called *categories*, *items* and *attributes*. Our Semantic Annotation of a Catalog with respect to a product ontology is based on the annotation of categories (called semantic entity in [1]) and is formally defined as follows. An annotation element is a tuple < *SE, AR, concept_description*> where *SE* is a semantic entity of the catalog; *concept_description* is a concept description of the *product ontology*; AR specifies the *Annotation Relation* which may hold between *SE* and *concept_description*; we consider *equivalence* (*AR_EQUIV*); *more general* (*AR_SUP*); *less general* (*AR_SUB*). Let us give some examples.

- (eBay:ClassicToys *AR_SUB* UNSPSC:Toys) this annotation declares that the entity eBay:ClassicToys is less general than the concept UNSPSC:Toys
- (eBay:ClassicToys *AR_SUP* UNSPSC:ToyTrains) this annotation expresses the fact that all instances of UNSPSC:ToyTrains would be classified in eBay:ClassicToys

3 Derived Mappings between Ontologized Catalogues

Assume now that several catalogues ontologized with respect to some standard ontologies. We now want to establish correspondences among two or more such catalogues, so that e.g. our purchasing department will be able to see and compare the offer of different suppliers for the same class of goods. The plan is to align the relevant parts of the reference product and services ontologies used to annotate the catalogues, and then to derive a map on the underlying catalogues.

Ontology alignment

The basic expression of alignment mapping for ontologies modeled with description logic formalisms involves the use of a semantic (logic) constructs or evolved frameworks to express the existence and properties of similarities and then mappings [4][7][10][12]. In this paper we use a somewhat simplified setup. Let O1 and O2 be ontologies. Then, an *entity alignment mapping* between entities E1 in O1 and E2 in O2 is a tuple < E1, AM_R, E2> where AM_R specifies the semantic relation which holds between E1 and E2: equivalence (AM_EQUIV), subClass (AM_SUBS) and superClass (AM_SUP). The above notation then reads "E1 is a AM_R of E2".

For example < UNSPSC:ToyTrains, AR_SUBS, ECLASS:Toys>

The mapping process

We are in a position now to establish mappings between our (ontologized) catalogues at last, and the reader should keep in mind the picture in Figure 1. The idea is that the mappings at the ontology level will actually induce mappings at the lower level.

Let's begin with a simple example. Given the eBay catalog, and another catalog that we call C2. Suppose that eBay has been ontologized with respect to UNSPSC, C2 has been ontologized using eCl@ss, and an alignment mapping has been established between UNSPSC and eCl@ss. If the following three facts have been established:

1. (eBay:ClassicToys AR_SUB UNSPSC:Toys)
2. (C2:SE1 AR_SUP ECLASS:Toys)
3. <ECLASS:Toys, AM_EQUIV UNSPSC:Toys>

Then, from 1. and 3. we can deduce: (eBay:ClassicToys AR_SUB ECLASS:Toys); And from 2. and 4. we conclude <C2:SE1, SUP, eBay:ClassicToys> (A), which establishes a mapping at the ontologized catalog level (where SUP denotes moreGeneral at the ontologized catalogues level). This is a *derived mapping* from the ontology alignment, realizing the picture in Figure 1. We should note now that if we had 2'. (C2:SE1 AR_SUB ECLASS:Toys), then our reasoning would collapse and we would not be able to assert any mapping at the ontologized catalogues level. This is a common occurrence, since in real life conditions there is no guarantee that all of the mappings at the ontology level will actually induce mappings at the lower level.

The type of mapping should also be considered. The statement (A) above declares that a certain entity in the C2 catalog includes all the eBay:ClassicToys. This is a true fact, but it is not obvious that it is the fact we want. For example, we might want to have a stronger or stricter property. This may come as a further deduction from other mappings, but it may not. In the latter case, those in charge of the mapping need to enhance the annotation of the catalogues, refine the ontology alignment and finally, if all else fails, force the desired mappings by hand

At this point the discriminating reader may wonder whether this process actually returns some mappings at the ontologized catalogues level. The answer is affirmative, at least in some reasonable circumstances. In fact, we can state the following

Theorem. Assume that O1 and O2 are reference ontologies ontologizing the catalogues C1 and C2 via annotations A1, A2 resp. For all entities E1 in C1 and E2 in C2 with annotations (C1:E1 AR_SUB O1:o1), (C2:E2 AR_SUB O2:o2), if we have the mapping <M1, O1:o1, AM_SUB, O2:o2> and O2:o2 is the image of C2:E2 via the annotation A2, then M1 translates into a mapping <T1, C1:E1, SUB, C2:E2>

The proof of this statement follows immediately from the unfolding of the definitions. This theorem shows that mappings at the ontologized catalog level are generated indeed, provided that we can map all the entities in our classification schemes into entities in the reference ontologies, which is mostly the case if the reference ontologies are worth their salt.

Next example shows how a property established in an ontology may propagate to the other ontology and both ontologized catalogues. Let O1, O2, be reference ontologies, and C1, C2, catalogues that have been ontologized with respect to O1, O2, E1i (i=1,2) entities in C1 and E2i (i=1,2) entities in C2, o1i (i=1,2) classes in O1 and o2i (i=1,2) classes in O2. Assume the following facts:

1. (C1:E1i AR_SUB O1:o1i) (i=1,2)
2. (C2:E2i AR_SUB O2:o2i) (i=1,2)
3. <M1i, O1:o1i, AM_SUB, O2:o2i> (i=1,2)
4. areDisjoint(O2:o21,O2:o22)

Then, a reasoner should be able to infer that E21 and E22 are disjoint, that O1:o11 and O1:o12 are disjoint, and finally that E11 and E12 are also disjoint. The nice outcome of this line of reasoning is therefore that any strong separation property established in O2 will propagate to O1 and both catalogues. This means that a comparison of ontologized catalogues can propagate qualifying properties and improve the quality of all the structures involved.

4 Conclusions and Future Work

This paper outlines the results of some scouting done in the area of effective and sustainable management of mappings among common industry tools like catalogues. While this exercise applies some general techniques in a specific context, it is suggestive of potential generalizations and difficulties to be tackled next.

The most interesting development should be to understand the relation between the mappings at the ontology level and derived mappings at the ontologized catalogues level in Figure 1, as modulations of the annotations of the catalogues using the ontologies.

Moreover, a more extensive approach including relations, instances, properties, rules, axioms, and constraints should be progressively pursued. This will enhance our

understanding of the properties that we should strive to identify a priori, in order to ease a forthcoming mapping process. More generally, the interplay of annotations and alignments could be investigated for general mappings between ontologies.
Finally, catalogues are one single area of interest. They are usually simply structured, yet large, occasionally idiosyncratic, evolving in time, reflecting real business needs. As such, they are a very relevant sandbox to try ideas for semantic applications. Eventually these techniques should migrate to other fields like EDI and general industrial and commercial operations of all kinds.

6 References

[1] Beneventano, D., Dahlem, N., El Haoum, S., Hahn, A., Montanari, D., and Reinelt, M., 2008. Ontology-driven Semantic Mapping, (I-ESA 2008) in Enterprise Interoperability III – Springer, Berlin, pp. 329-342.

[2] Beneventano, D., El Haoum, S., Montanari, D., 2007. Mapping of heterogeneous schemata, business structures, and terminologies, in DEXA Workshop on Database and Expert Systems Applications, IEEE Comper Society, pp. 412-418.

[3] D. Beneventano, D., Magnani, S., A 2004. Framework for the Classification and the Reclassification of Electronic Catalogs, 19th Annual ACM Symposium on Applied Computing Nicosia, Cyprus, March 14 -17.

[4] Choi, N., Song, I-Y., Han, H., 2006. A Survey of Ontology Mapping. In SIGMOD Record 35, Nr. 3.

[5] Corcho, O. and Gomez-Perez, A., 2001. Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings, Workshop on E-Business and Intelligent Web at the IJCAI2001 , Seattle, USA, August 5.

[6] eCl@ss web site: http://www.eclass-online.com/ (accessed on July 19, 2008).

[7] Euzenat, J., Mocan, A., Scharffe, F., 2008. Ontology alignments. In Hepp, M., De Leenheer, P., de Moor, A., and Sure, Y., Eds. 2008 Ontology Management. Springer. Semantic Web and Beyond: Computing for Human Experience.

[8] Hepp, M., 2006. Product and Service Ontologies: A Methodology for Deriving OWL ontologies from Industrial Categorization Standards. Int'l Journal on Semantic Web & Information Systems (IJSWIS), Vol. 2, No. 1, pp. 72-99.

[9] Klein, M., 2002. Web page on DAML+OIL and RDF Schema Representation of UNSPSC. http://www.cs.vu.nl/~mcaklein/unspsc/ (accessed on July 20, 2008).

[10] Ontology Alignment Evaluation Initiative (OAEI) web site: http://oaei.ontologymatching.org/ (accessed on July 21, 2008).

[11] RosettaNet http://www.rosettanet.org/cms/sites/RosettaNet/

[12] Shvaiko, P., Euzenat, J., 2005. A survey of schema-based matching approaches. Journal on Data Semantics 3730, pp. 146-171.

[13] UNSPSC web site: http://www.unspsc.org/ (accessed on July 19, 2008).

Towards Dialogue-Based Interactive Semantic Mediation in the Medical Domain

Daniel Sonntag

German Research Center for Artificial Intelligence
66123 Saarbrücken, Germany
sonntag@dfki.de

Abstract. We think of ontology matching as a dialogue-based interactive mediation process for which we propose a three stage model. A preliminary evaluation shows how we applied this method of eliciting input for ontology matching in the medical domain. Especially, we address the challenge how to use dialogue-based interactivity with the user to rate partial alignments between two ontologies.

Keywords: Dialogue Systems, Ontology Matching, Medical Image Retrieval

1 Introduction

One of the main goals of ontology integration is the interoperability of different ontologies. This means to allow to query different ontological databases for specific information, e.g., clinical images annotated with ontological metadata. At the best, the involved ontologies are already integrated. These approaches have been followed in large-scale Semantic Web projects, e.g., SmartWeb [1], Musing[1], and MESH[2]. However, many end user applications are dynamic and evolve over time. This means new information sources are to be added dynamically, which applies to information and knowledge retrieval applications in particular. In the case of knowledge retrieval from ontologies, this also means to align ontologies iteratively. Thus, application areas such as relational database integration, ontology merging, semantic web service composition, semantic peer-to-peer networks, and semantic query answering benefit from interactive semantic mediation with increasing intensity. In this paper, we identify dialogue-based interactive ontology matching as one of the largely unaddressed challenges in the area of semantic information integration.

Some early ontology merging/aligning systems (including the well-known PROMPT and Chimaera) adopt semi-automatic matching techniques through incremental interaction with users. Rather than focussing on the effectiveness of a further interactive matching approach, we describe a suitable dialogue-level integration of the interactive matching process by example. Dialogue-based interaction systems in the sense of conversational user interfaces stand for improved

[1] *http://www.musing.eu/*

[2] *http://www.mesh-ip.eu/*

efficiency and ease of use. For example, in the medical domain, radiologists in hospitals can pose clinically relevant questions such as *"Is there a different diagnosis for a lesion in our knowledge base or any other clinical resource available?"* The challenge is, of course, to find the relevant instance ontologies and to align them. For the interactive alignment process, we use the same dialogue shell as for the task-based dialogue, with state-of-the-art natural language processing methods combined in a Semantic Web framework [2, 3].

In this work, we count on dynamic models of the ontology integration process and try to model the meta dialogue for interactive ontology matching. Here, meta dialogue is to be understood as clarifications and questions that elicit user's feedback to support the ontology matching process. For example:[3]

1. **U:** "Show me the internal organs: lungs, liver, then spleen and colon."
2. **S:** Shows corresponding patient image data according to referral record.
3. **U:** "This lymph node here (+ pointing gesture) is enlarged; so *lymphadenopathy* (finding Radlex_Term_A). Are there any comparative cases in the hospital?"
4. **S:** "Sorry, there are no comparative cases, shall I continue to search for similar cases?"
5. **U:** "Ah okay."
6. **S:** "Unfortunately, external databases could not be matched appropriately. Is lymphadenopathy equivalent to *lymph node enlargement* (Radlex_Term_B) or *adenopathy* (NCI_Term_C) or *a kind of hematopoietic system finding* (NCI superconcept)?"

7. **U: "Yes, that could be said,"**
8. **U: "a pathophysiologic process."** (Medico now includes RadLex superconcepts for search.)
9. **S:** "I found comparative cases in an external knowledge base according to your feedback." Medico switches to the comparative records to help the radiologist in the differential diagnosis of the suspicious case, before the next organ (liver) is examined.
10. **U:** "Find similar liver lesions with the characteristics: hyper-intense and/or coarse texture ..."
11. **S:** Medico again displays the search results ranked by the similarity and matching of the medical terms that constrain the semantic search.

A useful and cooperative question answering dialogue in natural language would combine different topics, heterogeneous information sources, and user feedback on the matching process as meta dialogue. The example dialogue illustrates such a lifelike question answering dialogue; in this respect, utterance (6) is the meta level system question, and utterance (7) the user's interactive mapping feedback. It is to be mentioned that the system utterance (6) demands for a classification model that judges the accuracy of an ad hoc mapping[4]; the potential of the user feedback (7) is of course not limited to a singe correspondence which can be demonstrated by fixpoint alignment computation in similarity flooding; (8) shows user-initiative mapping information for possible supertypes.

[3] The potential application scenario (provided by Siemens AG in context of the THESEUS-Medico project) includes a radiologist which treats a lymphoma patient; the patient visits the doctor after chemotherapy for a follow-up CT examination. One of the radiologist's goals is to estimate the effectiveness of the administered medicine. In order to finish the reading/pathology, additional cases have to be taken into account for comparison, which we try to find by matching ontologies of different patient case databases.

[4] To our best knowledge, such a classification model has not yet been proposed in literature. We made good first experiences with a string-based model on the concept signs for complete mappings, where we computed the ratio of alignments with confidence value $t > 0.9$. However, this strategy is not robust in the case of partial mappings.

2 Dialogue-Based Interactive Matching Approach

The ontology matching problem can be addressed by several techniques (cf. [4] for example). Recent work in incremental interactive schema matching stressed that users are often annoyed by false positives [5]; advanced incremental visualisations have been developed (e.g., see [6]) to do better than calculate the set of correspondences in a single shot; cognitive support frameworks for ontology mapping really involve users [7]; a dialogue-based approach could make more use of partial mappings in addition, to increase the usability in dialogue scenarios where the primary task is different from the schema matching task itself. Our basic idea is as follows: Consider the methods that are required for interactive ontology mapping and evaluate the impact of dialogue-based user feedback in this process. While dialogue systems allow to obtain user feedback on semantic mediation questions (e.g., questions regarding new semantic mediation rules), incrementally working matching systems can use the feedback as further input for alignment improvement. In order to compute and post-process the alignments, we use the PhaseLibs library[5]. Subsequently, we focus on interactive ontology matching and dialogue-based interaction. Rather than focussing on the effectiveness of the interactive matching approach, we describe a suitable dialogue-level integration of the matching process by example. Our interactive ontology matching approach envisions the following three stages:

1. Compute a rudimentary partial mapping by a simple string-based method;
2. Ask the user to disambiguate some of the proposed mappings;
3. Use the resulting alignments as input for more complex algorithms.

What concern the first point, we hypothesise that the rudimentary mapping based on the concept and relation signs can be easily computed and obtained in dialogical reaction time (less than 3 seconds even for large ontologies); for second, user interactivity is provided by improving the automatically found correspondences through filtering the alignment. Concerning the third point, we employ similarity flooding, since it allows for input alignments and fixpoint computation in Phaselib's implementation following [8]. The interactive semantic mediation approach is depicted in figure 1. In order not to annoy the user, she is presented the difficult cases for disambiguation feedback only; thus we use the application dialogue shell basically for confirming or rejecting pre-considered alignments. The resulting alignments are then serialised as instances of an RDFS alignment format. Assuming that subsequent similarity computations successfully use the partial alignment inputs (to produce query-relevant partial alignment output), the proposed mediator can be said to be a light-weight but powerful approach to support incremental matching.

[5] See *http://phaselibs.opendfki.de*: This platform, for first, supports custom combinations of algorithms; for second, it is entirely written in Java which allows us to directly integrate the API into the dialogue shell; for third, the API supports individual modules and libraries for ontology adapters, similarity measures (e.g., string based, instance based, or graph based), and alignment generators.

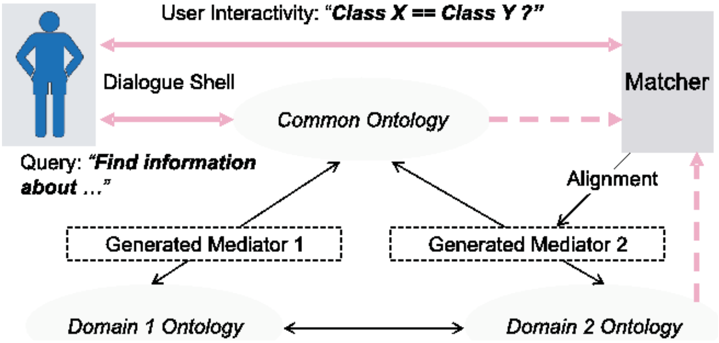


Fig. 1. Dialogue-based interactive semantic mediation approach

3 Conclusion and Future Work

We performed a series of preliminary experiments. Our datasets consisted of ontologies and alignment examples (manually annotated alignments for Radlex and NCI). For the first test in the medical domain, we annotated 50 alignments, 30 perfect positives and 20 perfect negatives.[6] This allows to compute a confusion matrix of the outcomes. In particular, in this domain the precision was 92% and recall 50% for simple string-based methods. (Corresponding concept names may differ substantially in their syntactic form.) Subsequently, the three best matches were taken as alignment input for similarity flooding after manually confirming their validity (which simulates positive user feedback). In subsequent tests, we compared the performance of similarity flooding (stage 3) with and without the initial alignments. Our experiments showed that, on average, the first stage of the matching execution (string-based matching) takes less than 5 percent of the end-to-end ontology matching execution time when similarity flooding is involved. In addition, the input alignments (confirmed by the simulated dialogue) allow to compute a complete mapping almost 10 times faster within a 30 seconds time frame;[7] a positive effect of partial mapping results with and without initial alignments could not yet be shown in terms of precision/accuracy.

The evaluation showed that for our test cases, interactive semantic mediation can be implemented by a simple string-based method (stage 1), to fulfill the requirements pertinent in the medical domain; the user dialogue was simulated by validating three matching inputs (stage 2). Since instance ontologies are hard to find for specific domains like medicine, non-instance based methods as described

[6] The radiologist's domain consists of many perfect matches according to an almost identical conceptual anatomy and disease model behind it. Unfortunately, this only concerns local concept structures; in addition, only few radiology experts can provide reliable alignments.

[7] It is to be mentioned that dataset-specific factors may heavily affect the total execution time as well as the percentage contribution to execution time when comparing the two different similarity flooding stages.

are welcome alternatives (stage 3). In future work, we are trying to provide evaluation methods to estimate the contribution of partial alignments input when the retrieval stage is more complex than simple name comparison, as is the case for most of our medical query patterns; user-confirmed perfect mappings can be used in simple name matching retrieval contexts with perfect precision, but this does not reflect the nature of real-world industrial requirements (in particular, where the user cannot be supposed to deliver a reliable judgement). Further, we are investigating techniques to better translate formal mapping uncertainties into appropriate dialogue-level questions for the radiologist and to address the general difficulty that users might not be able to provide helpful feedback in the course of a dialogue.

Acknowledgements. This research has been supported in part by the THESEUS Programme in the Core Technology Cluster WP4, which is funded by the German Federal Ministry of Economics and Technology (01MQ07016). The responsibility for this publication lies with the author.

References

1. Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Baumann, S., Vembu, S., Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Loos, B., Zorn, H.P., Micelli, V., Porzel, R., Schmidt, C., Weiten, M., Burkhardt, F., Zhou, J.: DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO). Web Semant. **5**(3) (2007) 156–174
2. Reithinger, N., Sonntag, D.: An integration framework for a mobile multimodal dialogue system accessing the Semantic Web. In: Proceedings of Interspeech'05. (2005)
3. Sonntag, D., Engel, R., Herzog, G., Pfalzgraf, A., Pfleger, N., Romanelli, M., Reithinger, N.: SmartWeb Handheld—Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services. In Huang, T.S., Nijholt, A., Pantic, M., Pentland, A., eds.: Artifical Intelligence for Human Computing. Volume 4451 of Lecture Notes in Computer Science., Springer (2007) 272–295
4. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (2007)
5. Bernstein, P.A., Melnik, S., Churchill, J.E.: Incremental schema matching. In: VLDB '06: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment (2006) 1167–1170
6. Robertson, G.G., Czerwinski, M.P., Churchill, J.E.: Visualization of mappings between schemas. In: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2005) 431–439
7. Falconer, S.M., Noy, N., Storey, M.A.D.: Towards understanding the needs of cognitive support for ontology mapping. In Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M., eds.: Ontology Matching. Volume 225 of CEUR Workshop Proceedings., CEUR-WS.org (2006)
8. Melnik, S., Garcia-Molina, H., Rahm, E.: 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, Washington, DC, USA, IEEE Computer Society (2002) 117–128

Towards Ontology Interoperability through Conceptual Groundings

Stefan Dietze, John Domingue
Knowledge Media Institute,
The Open University,
MK7 6AA, Milton Keynes, UK
{s.dietze, j.b.domingue}@open.ac.uk

Abstract. The widespread use of ontologies raises the need to resolve heterogeneities between distinct conceptualisations in order to support interoperability. The aim of *ontology mapping* is, to establish formal relations between a set of knowledge entities which represent the same or a similar meaning in distinct ontologies. Whereas the symbolic approach of established SW representation standards – based on first-order logic and syllogistic reasoning – does not implicitly represent similarity relationships, the ontology mapping task strongly relies on identifying *semantic similarities*. However, while concept representations across distinct ontologies hardly equal another, manually or even semi-automatically identifying similarity relationships is costly. *Conceptual Spaces (CS)* enable the representation of concepts as vector spaces which implicitly carry similarity information. But CS provide neither an implicit representational mechanism nor a means to represent arbitrary relations between concepts or instances. In order to overcome these issues, we propose a hybrid knowledge representation approach which extends first-order logic ontologies with a conceptual grounding through a set of CS-based representations. Consequently, semantic similarity between instances – represented as members in CS – is indicated by means of distance metrics. Hence, automatic similarity-detection between instances across distinct ontologies is supported in order to facilitate ontology mapping.

Keywords: Semantic Web, Ontology Mapping, Conceptual Spaces, Interoperability.

1 Introduction

The widespread use of ontologies - formal specifications of shared conceptualisations [10] - together with the increasing availability of representations of overlapping domains of interest, raises the need to resolve heterogeneities [12][14] by completely or partially mapping between different ontologies. With respect to [2][17], we define *ontology mapping* as the process of defining formal relations between knowledge entities which represent the same or a similar semantic meaning in distinct ontologies [6][19]. In that, ontology mapping strongly relies on identifying *similarities* [1] between entities across different ontologies. However, with respect to this goal, several issues have to be taken into account. The symbolic approach - i.e. describing symbols by using other symbols, without a grounding in the real world - of established representation standards such as OWL[1] or RDF-S[2] which are based on first-order logic (FOL) and syllogistic reasoning [8] leads to ambiguity issues and

[1] http://www.w3.org/OWL/

[2] http://www.w3.org/RDFS/

does not entail meaningfulness, since meaning requires both the definition of a terminology in terms of a logical structure (using symbols) and grounding of symbols to a conceptual level [3][16]. Therefore, concept representations across distinct ontologies – even those representing the same real-world entities - hardly equal another, since similarity is not an implicit notion carried within ontological representations. But manual or semi-automatic identification of similarity relationships – based on linguistic or structural similarities across ontologies [13][7][9] – is costly. Consequently, representational facilities, enabling to implicitly describe similarities across ontologies are required in order to support ontology interoperability.

Conceptual Spaces (CS) [8] follow a theory of describing entities at the conceptual level in terms of their natural characteristics similar to natural human cognition in order to avoid the symbol grounding issue [3][16]. In that, CS consider the representation of concepts as vector spaces which are defined through a set of quality dimensions. Describing instances as vectors enables the automatic calculation of their semantic similarity by means of spatial distance metrics, in contrast to the costly representation of similarities through symbolic representations. However, several issues still have to be considered when applying CS. For instance, CS do not explicitly prescribe any applicable representation method. Moreover, CS provide no means to represent arbitrary relations between concepts or instances, such as part-of relations. In order to overcome the issues introduced above, we propose a two-fold knowledge representation approach which extends FOL ontologies with a conceptual grounding by refining individual symbolic concept representations as particular CS. Consequently, similarity becomes an implicit notion of the representation itself, instead of relying on manual or semi-automatic similarity detection approaches.

2 Conceptual Groundings for Ontological Concepts

With respect to the aforementioned issues, we argue that basing knowledge models on just one theory alone might not be sufficient. Therefore, we propose a two-fold representational approach – combining FOL ontologies with corresponding representations based on CS – to enable similarity-based reasoning across ontologies. In that, we consider the representation of a set of *n* concepts *C* of an ontology *O* through a set of *n* Conceptual Spaces *CS*. Hence, instances of concepts are represented as members in the respective CS. While still benefiting from implicit similarity information within a CS, our hybrid approach allows overcoming CS-related issues by maintaining the advantages of FOL-based knowledge representations. In order to be able to represent ontological concepts within CS, we formalised the CS model into an ontology, represented through OCML [15]. Hence, a CS can simply be instantiated in order to represent a particular concept.

Referring to [8][18], we formalise a CS as a vector space defined through quality dimensions d_i of *CS*. Each dimension is associated with a certain metric scale, e.g. ratio, interval or ordinal scale. To reflect the impact of a specific quality dimension on the entire CS, we consider a prominence value *p* for each dimension [8]. Therefore, a CS is defined by $CS^n = \{(p_1 d_1, p_2 d_2, ..., p_n d_n) | d_i \in CS, p_i \in P\}$, where *P* is the set of real numbers. However, the usage context, purpose and domain of a particular CS strongly influence the ranking of its quality dimensions what supports our position of describing distinct CS explicitly for individual concepts. Please note that we do not distinguish between dimensions and domains [8] but enable dimensions to be detailed further in terms of subspaces. Hence, a dimension within one space may be defined through another CS by using further dimensions [18]. In this way, a CS may be composed of several subspaces, and consequently, the description granularity can be refined gradually. Dimensions may be correlated. Information about correlation is expressed through axioms related to a specific quality dimension instance.

A particular member *M* – representing a particular instance – in the CS is described through valued dimension vectors v_i like $M^n = \{(v_1, v_2, ..., v_n) | v_i \in M\}$. With respect to [18], we define the semantic similarity between two members of a space as a function of the Euclidean distance between the points representing each of the members. However, we would like to point out that different distance metrics, such as the Taxicab or Manhattan distance [11], could be considered, dependent on the nature and purpose of the CS. Given a CS definition *CS* and two members *V* and *U*, defined by vectors $v_0, v_1, ..., v_n$ and $u_1, u_2, ..., u_n$ within *CS*, the distance between *V* and *U* can be calculated as $dist(u,v) = \sqrt{\sum_{i=1}^{n} p_i((\frac{u_i - \bar{u}}{s_u}) - (\frac{v_i - \bar{v}}{s_v}))^2}$ where $\bar{u}$ is the mean of a dataset *U* and s_u is the standard deviation from *U*. The formula above already considers the so-called Z-transformation or standardization [4] which facilitates the standardization of distinct measurement scales in order to enable the calculation of distances in a multi-dimensional and multi-metric space.

Representing Ontological Concepts through Conceptual Spaces

The derivation of an appropriate space CS_i to represent a particular concept C_i of a given ontology *O* is understood a non-trivial task which primarily implies the creation of a CS instance which most appropriately represents the real-world entity represented by the symbolic concept representation. We foresee a transformation procedure consisting of the following steps:

S1. Representing concept properties pc_{ij} of C_i as dimensions d_{ij} of CS_i.

S2. Assignment of metrics to each quality dimension d_{ij}.

S3. Assignment of prominence values p_{ij} to each quality dimension d_{ij}.

S4. Representing instances I_{ik} of C_i as members in CS_i.

A specific CS is instantiated by applying a transformation function which is aimed at instantiating all elements of a CS *(S1 – S3*). *S1* aims at representing each concept property pc_{ij} of C_i as a particular dimension instance d_{ij} together with a corresponding prominence p_{ij} of a resulting space CS_i:

$$trans: \{(pc_{i1}, pc_{i2}, ..., pc_{in}) | pc_{ij} \in PC\} \Rightarrow \{(p_{i1}d_{i1}, p_{i2}d_{i2}, ..., p_{in}d_{in}) | d_{ij} \in CS_i, p_{ij} \in P\}$$

Please note that we particularly distinguish between data type properties and relations. While the latter represent relations between concepts, these are not represented as dimensions since such dimensions would refer to a range of concepts (instances) instead of quantified metrics, as required by *S2*. In the case of relations, we propose to maintain the relationships represented within the original ontology *O* without representing these within the resulting CS_i. In that, the complexity of CS_i is reduced to enable the maintainability of the spatial distance as appropriate similarity measure. *S2* aims at the assignment of metric scales (interval scale, ratio scale, nominal scale), while *S3* is aimed at assigning a prominence value p_{ij} to each dimension d_{ij}. Prominence values should be chosen from a predefined value range, such as 0...1. With respect to *S4*, one has to represent all instances I_{ki} of a concept C_i as member instances in the created space CS_i. This is achieved by transforming all instantiated properties pi_{ikl} of I_{ik} as valued vectors in CS_i.

$$trans : \{(pi_{ik1}, pi_{ik2}, \ldots, pi_{ikn}) | pi_{ikl} \in PI_l\} \Rightarrow \{(v_{ik1}, v_{ik2}, \ldots, v_{ikn}) | v_{ikl} \in M_{ik}\}$$

Hence, given a particular CS, representing instances as members becomes just a matter of assigning specific measurements to the dimensions of the CS. In order to represent all concepts C_i of a given ontology *O*, the transformation function consisting of the steps *S1-S4* has to be repeated iteratively for all C_i which are element of *O.* The accomplishment of the proposed procedure results in a set of CS instances which each refine a particular concept together with a set of member instances which each refine a particular instance.

3 Conclusion

In order to facilitate ontology mapping, we proposed a hybrid representation approach based on a combination of FOL ontologies and multiple concept representations in individual CS. Representing concepts following the CS theory enables representation of instances as vectors in a respective CS and consequently, the automatic computation of similarities by means of spatial distances. A CS-based representation is supported through a dedicated CS formalisation, i.e. a CS ontology, and a formal method on how to derive CS representations for individual concepts. Within proof-of-concept prototype applications, e.g. [5], an OCML [15] representation of the proposed hybrid representational model was utilized to validate the applicability of the approach. Following our two-fold representational approach supports implicit representation of similarities across heterogeneous ontologies, and consequently, provides a means to facilitate ontology mapping. Moreover, our approach overcomes certain individual issues posed by each of the two approaches. Whereas traditional ontology mapping methodologies rely on mechanisms to semi-automatically detect similarities at the concept and the instance level, our approach just requires a common agreement at the concept level since similarity information at the instance level is implicitly defined.

However, the authors are aware that our approach requires a considerable amount of additional effort to establish CS-based representations. Future work has to investigate this effort in order to further evaluate the potential contribution of the approach proposed here. Moreover, further issues related to CS-based knowledge representations still remain. For instance, whereas defining instances, i.e. vectors, within a given CS appears to be a straightforward process, the definition of the CS itself is not trivial at all and dependent on subjective perspectives. With regard to this, CS do not fully solve the symbol grounding issue but to shift it from the process of describing instances to the definition of a CS. Nevertheless, distance calculation relies on the fact that resources are described in equivalent (or mapped) geometrical spaces. However, we would like to point out that the increasing usage of upper level ontologies and the progressive reuse of ontologies, particularly in loosely coupled organisational environments, leads to an increased sharing of ontologies at the concept level. As a result, our proposed hybrid representational model becomes increasingly applicable by further enabling similarity-computation at the instance-level towards the vision of interoperable ontologies.

4 References

[1] Bisson, G. (1995). Why and how to define a similarity measure for object based representation systems. Towards Very Large Knowledge Bases, pages 236–246, 1995.

[2] Choi, N., Song, I., and Han, H. (2006), A survey on ontology mapping, SIGMOD Rec., Vol. 35, No. 3. (September 2006), pp. 34-41.

[3] Cregan, A. (2007), Symbol Grounding for the Semantic Web. 4th European Semantic Web Conference 2007, Innsbruck, Austria.

[4] Devore, J., and Peck, R. (2001), *Statistics - The Exploration and Analysis of Data*, 4th ed. Pacific Grove, CA: Duxbury, 2001.

[5] Dietze, S., Gugliotta, A., Domingue, J., (2008) Conceptual Situation Spaces for Situation-Driven Processes. 5th European Semantic Web Conference (ESWC), Tenerife, Spain.

[6] Ehrig, M, Sure, Y. (2004), Ontology Mapping - An Integrated Approach, in Proceedings of ESWS, 2004.

[7] Euzenat, J., Guegan, P., and Valtchev, P. OLA in the OAEI 2005 Alignment Contest. K-Cap 2005 Workshop on Integrating Ontologies 2005, 97-102.

[8] Gärdenfors, P. (2004), How to make the semantic web more semantic. In A.C. Vieu and L. Varzi, editors, Formal Ontology in Information Systems, pages 19–36. IOS Press, 2004.

[9] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. S-Match: An Algorithm and an Implementation of Semantic Matching. ESWS 2004, 61-75

[10] Gruber, T. R., Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, Vol. 43, Issues 4-5, November 1995, pp. 907-928.

[11] Krause, E. F. (1987). Taxicab Geometry. Dover.

[12] Le, B.T., Dieng-Kuntz, R., and Gandon, F. On Ontology Matching Problems - for Building a Corporate Semantic Web in a Multi-Communities Organization. ICEIS (4) 2004, 236-243.

[13] Mitra, P., Noy, F. N., Jaiswals, A. (2005), OMEN: A Probabilistic Ontology Mapping Tool, International Semantic Web Conference 2005.

[14] Moncan, A., Cimpian, E., Kerrigan, M., Formal Model for Ontology Mapping Creation, in I. Cruz et al. (Eds.): ISWC 2006, LNCS 4273, pp. 459–472, 2006. Springer-Verlag Berlin Heidelberg 2006.

[15] Motta, E. (1998). An Overview of the OCML Modelling Language, the 8th Workshop on Methods and Languages, 1998.

[16] Nosofsky, R. (1992), Similarity, scaling and cognitive process models, Annual Review of Psychology 43, pp. 25- 53, (1992).

[17] Pinto, S., H., Gomez-Perez, A., Martins, J. P. (1999), Some Issues on Ontology Integration, In Proc. of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends, 1999.

[18] Raubal, M. (2004) Formalizing Conceptual Spaces. in: A. Varzi and L. Vieu (Eds.), Formal Ontology in Information Systems, Proceedings of the Third International Conference (FOIS 2004). Frontiers in Artificial Intelligence and Applications 114, pp. 153-164, IOS Press, Amsterdam, NL.

[19] Xiaomeng Su. (2002). A text categorization perspective for ontology mapping, Technical report, Department of Computer and Information Science, Norwegian University of Science and Technology, Norway, 2002.

ISWC2004

ISWC 2008

The 7th International Semantic Web Conference
October 26 – 30, 2008
Congress Center, Karlsruhe, Germany