Description of alignment implementation and benchmarking results

Heiner Stuckenschmidt, Marc Ehrig, Jérôme Euzenat, Andreas Hess, Willem Robert van Hage, Wei Hu, Ningsheng Jian, Gong Chen, Yuzhong Qu, George Stoilos, et al.

D2.2.4: Alignment implementation and benchmarking results

Coordinator: Heiner Stuckenschmidt (University of Mannheim)
Marc Ehrig (Universität Karlruhe),
Jérôme Euzenat (INRIA Rhône-Alpes)
Andreas Hess, Willem Robert van Hage (Vrije Universiteit Amsterdam)
Wei Hu, Ningsheng Jian, Gong Cheng and Yuzhong Qu (Southeast University China)
George Stoilos, George Stamou (ITI-Certh)
Umberto Straccia (ISTI-CNR)
Vojtech Svatek (University of Economics, Prague)
Raphaël Troncy (CWI Amsterdam)
Petko Valtchev (Université de Montréal),
Mikalaï Yatskevich (Universita Trento)

Abstract.
This deliverable presents the evaluation campaign carried out in 2005 and the improvement participants to these campaign and others have to their systems. We draw lessons from this work and proposes improvements for future campaigns.
Keyword list: ontology matching, ontology alignment, ontology mapping, evaluation, benchmarking, contest, performance measure.

Document Identifier	KWEB/2005/D2.2.4/v1.1
Project	KWEB EU-IST-2004-507482
Version	v1.1
Date	January 6, 2006
State	draft
Distribution	public

Knowledge Web Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2004-507482.

University of Innsbruck (UIBK) - Coordinator
Institute of Computer Science
Technikerstrasse 13
A-6020 Innsbruck
Austria
Contact person: Dieter Fensel
E-mail address: dieter.fensel@uibk.ac.at

École Polytechnique Fédérale de Lausanne (EPFL)
Computer Science Department
Swiss Federal Institute of Technology
IN (Ecublens), CH-1015 Lausanne
Switzerland
Contact person: Boi Faltings
E-mail address: boi.faltings@epfl.ch

France Telecom (FT)
4 Rue du Clos Courtel
35512 Cesson Sévigné
France. PO Box 91226
Contact person : Alain Leger
E-mail address: alain.leger@rd.francetelecom.com

Freie Universität Berlin (FU Berlin)
Takustrasse 9
14195 Berlin
Germany
Contact person: Robert Tolksdorf
E-mail address: tolk@inf.fu-berlin.de

Free University of Bozen-Bolzano (FUB)
Piazza Domenicani 3
39100 Bolzano
Italy
Contact person: Enrico Franconi
E-mail address: franconi@inf.unibz.it

Institut National de Recherche en Informatique et en Automatique (INRIA)
ZIRST - 655 avenue de l'Europe -
Montbonnot Saint Martin
38334 Saint-Ismier
France
Contact person: Jérôme Euzenat
E-mail address: Jerome.Euzenat@inrialpes.fr

Centre for Research and Technology Hellas / Informatics and Telematics Institute (ITI-CERTH)
1st km Thermi - Panorama road
57001 Thermi-Thessaloniki
Greece. Po Box 361
Contact person: Michael G. Strintzis
E-mail address: strintzi@iti.gr

Learning Lab Lower Saxony (L3S)
Expo Plaza 1
30539 Hannover
Germany
Contact person: Wolfgang Nejdl
E-mail address: nejdl@learninglab.de

National University of Ireland Galway (NUIG)
National University of Ireland
Science and Technology Building
University Road
Galway
Ireland
Contact person: Christoph Bussler
E-mail address: chris.bussler@deri.ie

The Open University (OU)
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA
United Kingdom
Contact person: Enrico Motta
E-mail address: e.motta@open.ac.uk

Universidad Politécnica de Madrid (UPM)
Campus de Montegancedo sn
28660 Boadilla del Monte
Spain
Contact person: Asunción Gómez Pérez
E-mail address: asun@fi.upm.es

University of Karlsruhe (UKARL)
Institut für Angewandte Informatik und Formale
Beschreibungsverfahren - AIFB
Universität Karlsruhe
D-76128 Karlsruhe
Germany
Contact person: Rudi Studer
E-mail address: studer@aifb.uni-karlsruhe.de

University of Liverpool (UniLiv)
Chadwick Building, Peach Street
L697ZF Liverpool
United Kingdom
Contact person: Michael Wooldridge
E-mail address: M.J.Wooldridge@csc.liv.ac.uk

University of Manchester (UoM)
Room 2.32. Kilburn Building, Department of Computer
Science, University of Manchester, Oxford Road
Manchester, M13 9PL
United Kingdom
Contact person: Carole Goble
E-mail address: carole@cs.man.ac.uk

University of Sheffield (USFD)
Regent Court, 211 Portobello street
S14DP Sheffield
United Kingdom
Contact person: Hamish Cunningham
E-mail address: hamish@dcs.shef.ac.uk

University of Trento (UniTn)
Via Sommarive 14
38050 Trento
Italy
Contact person: Fausto Giunchiglia
E-mail address: fausto@dit.unitn.it

Vrije Universiteit Amsterdam (VUA)
De Boelelaan 1081a
1081HV. Amsterdam
The Netherlands
Contact person: Frank van Harmelen
E-mail address: Frank.van.Harmelen@cs.vu.nl

Vrije Universiteit Brussel (VUB)
Pleinlaan 2, Building G10
1050 Brussels
Belgium
Contact person: Robert Meersman
E-mail address: robert.meersman@vub.ac.be

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to writing parts of this document:

Centre for Research and Technology Hellas
École Polytechnique Fédérale de Lausanne
Free University of Bozen-Bolzano
Institut National de Recherche en Informatique et en Automatique
National University of Ireland Galway
Universidad Politécnica de Madrid
University of Innsbruck
University of Economy Prague
University of Karlsruhe
University of Manchester
University of Mannheim
University of Sheffield
University of Trento
Vrije Universiteit Amsterdam
Vrije Universiteit Brussel

Changes

Version	Date	Author	Changes
0.1	12.07.2005	Jérôme Euzenat	creation
0.2	10.10.2005	Jérôme Euzenat	filled with OAEI material
0.5	5.12.2005	Heiner Stuckenschmidt	added Part I
0.6	5.12.2005	Heiner Stuckenschmidt	added Chapters 10 and 11
0.7	9.12.2005	Jérôme Euzenat	added OLA section, improved Chapter 8
0.9	12.12.2005	Heiner Stuckenschmidt	Executive summary, Conclusions
1.0	13.12.2005	Heiner Stuckenschmidt	Minor Fixes
1.1	13.12.2005	Jérôme Euzenat	Minor Fixes

Executive Summary

Deliverable 2.2.3 provided a survey of the state of the art in ontology matching and alignment prior to the start of the KnowledgeWeb project. One of the central goals of workpackage 2.2 is to advance this state of the art in a measurable way.

This deliverable reports on results towards this goal focussing on two issues:

1. Improvements in the area of methods and tools for the automatic alignment of ontologies
2. Methodological and practical aspects of evaluating and comparing alignment tools

We start with a description of current advances in automatic alignment technology. In particular, we describe methods that are improvements of alignment methods reported in deliverable 2.2.3. In addition, we describe a number of new alignment methods that have been developed since the publication of D2.2.3.

Systematic evaluation is essential for assessing the state of the art in ontology alignment and provides the basis for measuring advances in the field. In the context of the KnowledgeWeb project such a systematic evaluation is carried out in the form of an ontology alignment evaluation initiative that organizes an alignment challenge in which alignment tools compete on predefined alignment problems. The results of the different tools are evaluated based on a well-defined evaluation methodology which is described in deliverable 2.2.3.

In this deliverable, we present and discuss the results of the latest alignment challenge and draw conclusions about recent achievements and open problems. We identify three main problems that have to be addressed in more details:

1. Standard quality measures like precision and recall known from information retrieval do not adequately address the needs of the ontology alignment problem.
2. The generation of high quality reference alignments to compare automatically generated alignments again is an open problem when it comes to realistic alignment tasks.
3. The selection of test data sets is critical as certain data sets only cover certain aspects of the general alignment task.

These identified problems are addressed in the last part of the deliverable. We present a discussion of alternative quality measures for assessing automatically generated ontology mappings. We also discuss the problem of generating reference alignments based on shared instances for different types of conceptual structures (in particular simple classifications and thesauri). Finally, we discuss different possible test data sets to be used in future alignment challenges.

In summary, this report shows that there is progress on both, the development of alignment techniques and strategies for evaluating alignment. It also shows that more work is needed especially on the problem of designing and using benchmarks. This particular problem has to be

addressed in the context of industrial applications. In particular this question has to be addressed in the upcoming deliverable D1.2.1 on the utility of merging and alignment tools.

Contents

Part I

Improvement of Matching Algorithms

Chapter 1

The Dublin Algorithm for Ontology Alignment

Most mapping algorithms adhere to a simple structure: an initial calculation of an intrinsic similarity measure is followed by an iterative calculation of an extrinsic (structural) measure, before finally the mappings are derived from the pairwise similarities. Our algorithm follows this common structure, too. However, there are two features which make it distinct from other algorithms that we are aware of. First, we compute the structural similarity by using a feature vector representation of each concept. Section 1.2 describes the details. Second, the way how the similarities are transformed into mappings differs from most current approaches. While Melnik et al. in [Melnik *et al.*, 2002] propose to compute either a stable marriage or the maximum weighted matching over a bipartite graph that represents the pairwise similarities of concepts, it seems that most newer ontology mapping algorithms do not do this (e.g. Ehrig and Staab use a simple greedy approach in [Ehrig and Sure, 2004]). In section 1.3.1 we describe how these two well-known graph algorithms can be used.

1.1 Computing Intrinsic Similarity

We use URIs, labels, comments and text from individuals and property values as text sources. In our implementation, we use distance metrics from the well-known SecondString library[1] as intrinsic similarity measures. We used a version of Levenshtein edit distance [Levenshtein, 1966] that is scaled to the range $[0, 1]$ for comparing labels and local names. We used a soft-token metric for comparing comments and instance data. To determine the overall intrinsic similarity between two concepts, we use the maximum of these metrics. To avoid overemphasizing small similarities, we disregard similarities that are smaller than a threshold of 0.4 and map similarities greater than 0.4 to the full range $[0, 1]$.

1.2 Computing Extrinsic Similarity

To compute the extrinsic similarity, we use a vector representation $\vec{\mathrm{de}}(v)$ for each entity and then compute the similarities between these vectors. To formally define the extrinsic feature vector, we

[1] `http://secondstring.sourceforge.net/`, see also [Cohen *et al.*, 2003]

first introduce a function that computes all entities that are connected to an entity v by a relation l.

Definition 1. *We define a function from the set of vertices and the set of labels L to the power set of vertices so that for a given vertex the function finds all vertices adjacent through an arc with a given label:*

$$\mathrm{rel} : V \times L \to 2^V$$

Let $G = (V, A)$ be a digraph with the set of vertices V and labelled arcs A as a set of ordered triples $(v, w, l) \in V \times W \times L$. Then we define:

$$\mathrm{rel}(v, l) = \{x | v, x \in V \wedge (v, x, l) \in A\}$$

The definition of $\mathrm{rel}' : V' \times L \to 2^{V'}$ is analogous.

Next, as an intermediate step to our extrinsic feature vector function, we define a *dynamic intrinsic* feature vector function as a vector representation of all similarities between an entity v and all entities $v' \in V'$. Dynamic intrinsic means that these features are inherent to an entity, but they are dynamic in the sense that their value can change as we get more information about that entity and can thus make a better prediction about the similarities between this and other entities. Note that the dynamic intrinsic features are what we want to compute. In particular, this means that the dynamic intrinsic features are initially unknown.

Definition 2. *We define a dynamic intrinsic feature vector function as:*

$$\vec{\mathrm{di}} : V \to \mathbb{R}^{|V'|}$$

Analogous to the matrix representation of a graph, we impose an arbitrary total order on V' and denote the first element of V' as v'_0 and the subsequent elements as v'_n for all $n < |V'|$. Then we define $\vec{\mathrm{di}}$ as follows:

$$\vec{\mathrm{di}}(v) = [\mathrm{sim}(v, v'_0), \mathrm{sim}(v, v'_1), \ldots, \mathrm{sim}(v, v'_{|V'|-1})]$$

Dynamic extrinsic features are dynamic intrinsic features of related entities:

Definition 3. *We define a dynamic extrinsic feature vector function as:*

$$\vec{\mathrm{de}} : V \to \mathbb{R}^{|V'|}$$

Assuming a commutative and associative operator $\oplus$ on $\mathbb{R}^d$ and a function rel as per definition 1, we define $\vec{\mathrm{de}}(v)$ as some combination $\oplus$ of the dynamic intrinsic features $\vec{\mathrm{di}}(x)$ (see definition 2) of all related entities $x \in \mathrm{rel}(v)$.

$$\vec{\mathrm{de}}(v) = \bigoplus_{x \in \mathrm{rel}(v)} \vec{\mathrm{di}}(x)$$

Note that the elements in $\vec{\mathrm{de}}(v)$ are based on the relations of $v \in V$, but correspond to vertices in V'. In order to compute an extrinsic similarity between v and some v', we have to define an extrinsic feature vector for v' that is based on the relations of $v' \in V'$.

Definition 4. *We define an extrinsic feature vector function as:*

$$\vec{\mathrm{de}}' : V' \to \mathbb{R}^{|V'|}$$

Based on the total order on V' from definition 2, we define that each element i in $\vec{\mathrm{de}}'$ is 1, if $v'_i \in \mathrm{rel}(v')$ and 0 otherwise.

Algorithm 1 Iterative Similarity Calculation

for $v \in V$ **do**
  $\vec{\mathrm{di}}_{\mathrm{int}}(v) \leftarrow [\mathrm{sim}_{\mathrm{int}}(v, v'_0), \mathrm{sim}_{\mathrm{int}}(v, v'_1), \ldots, \mathrm{sim}_{\mathrm{int}}(v, v'_{|V'|-1})]$
end for
$\vec{\mathrm{de}}(v) \leftarrow \bigoplus_{x \in \mathrm{rel}(v)} \vec{\mathrm{di}}_{\mathrm{int}}(x)$ {Initially, use intrinsic similarity only}
for a fixed number of iterations **do**
  for $v \in V$ **do**
    $\vec{\mathrm{di}}_{\mathrm{ext}}(v) \leftarrow [\mathrm{sim}_{\mathrm{ext}}(v, v'_0), \mathrm{sim}_{\mathrm{ext}}(v, v'_1), \ldots, \mathrm{sim}_{\mathrm{ext}}(v, v'_{|V'|-1})]$
    $\vec{\mathrm{di}}(v) \leftarrow \vec{\mathrm{di}}_{\mathrm{int}}(v) \otimes \vec{\mathrm{di}}_{\mathrm{ext}}(v)$ {Combine intrinsic and extrinsic similarity}
  end for
  $\vec{\mathrm{de}}(v) \leftarrow \bigoplus_{x \in \mathrm{rel}(v)} \vec{\mathrm{di}}(x)$
end forreturn $\forall v \in V : \vec{\mathrm{di}}(v)$

Given definitions 3 and 4 we can now easily define an extrinsic similarity function $\mathrm{sim}_{\mathrm{ext}}(v, v')$ based on the similarity between the vectors $\vec{\mathrm{de}}(v)$ and $\vec{\mathrm{de}}'(v')$. A common similarity measure for two vectors is the dot product, but it is usually better to normalize the similarity measure using the well-known cosine, Dice, Jaccard or overlap coefficients, which are widely used in information retrieval, e.g. [van Rijsbergen, 1979] or [Salton, 1989].

The similarities based on the extrinsic feature vectors are not symmetric. Since the feature vector is based on the best mapping for each concept, the fact that v maps to v' does not necessarily mean that the best mapping for v' is v, if the overall similarity $\mathrm{sim}(v, v')$ is greater than the similarity of v to all other $x' \in V'$ but less than the similarity $\mathrm{sim}(v', x)$ of v' to some $x \in V$.

1.3 Iterative Algorithm

Algorithm 1 formally specifies the iterative method of calculating the overall similarity. We are not restricted to computing $\mathrm{sim}(v, v')$, calculating $\mathrm{sim}(v', v)$ is analogous. Recall that because of the way the extrinsic similarity is defined they are not necessarily equal. The next section explains a way to exploit this asymmetry.

This algorithm is in fact very similar to the supervised learning algorithm that we presented in [Heß and Kushmerick, 2004] and could be seen as a generalization thereof. For that reason it is straightforward to incorporate background knowledge, e.g. a mapping to a third ontology that is known a priori, if we substitute a machine learning algorithm instead of a string distance metric. We will explore this possibility in future work.

1.3.1 Postprocessing Steps

Once we have computed the overall similarities, we have to compute the actual one-to-one mapping. This is the problem of finding a matching in a bipartite graph. A bipartite graph $B = (V + V', E)$ is a graph where the nodes can be split in two groups such that every edge connects two nodes from both partitions. Every similarity that has been calculated in the previous step corresponds to a weighted edge in such a bipartite graph. A matching M in a graph is a set of edges such that no node is incident to more than one edge. In our setting this corresponds to

a one-to-one mapping: For every instance in one ontology we want to find one instance in the other ontology. M is called maximum-weighted, if there is no other matching where the sum of all edge weights in the matching is bigger. M is called a stable marriage, if there are no nodes $v \in V$ and $v' \in V'$ such that the edge between v and v' in B is not in M, but has a higher weight than the edges in M that are incident in v and v'. We used the Gale/Shapley algorithm [Gale and Shapley, 1962] to compute stable marriages and Munkres' algorithm [Munkres, 1957] (also referred to as the Hungarian algorithm) to compute maximum-weighted matchings.

The mappings submitted to the OAEI evaluation were computed with a fixed number of 5 iterations for the similarity calculation and using Munkres' algorithm to compute a maximum-weighted matching, which performed better than a setup with a stable marriage.

Chapter 2

oMAP: An Implemented Framework for Automatically Aligning OWL Ontologies

Ontologies are usually seen as a solution to data heterogeneity on the web [Euzenat and Valtchev, 2004]. An ontology is a way of describing the world: it allows to determine what kinds of things there are in the world, their characteristics, the relationships between them and more complex axioms. Since a lot of efforts are deployed to provide hands-on support for developers of Semantic Web applications, with the online publishing of "best practices", it is expected now that more and more ontologies covering partially the same subjects will be available on the web. Indeed, this is already true for numerous complex domains such that the medical or the multimedia domain. In such a case, some entities can be given different names or simply be defined in different ways or in different languages. The semantic interoperability has then to be grounded in ontology reconciliation. The underlying problem is often called the "ontology alignment" problem [Euzenat and Valtchev, 2004].

We focus here on ontologies described in the same knowledge representation language (OWL) and we propose a general framework named *oMAP* that aims to automatically align two OWL ontologies. *oMAP* [Straccia and Troncy, 2005b, Straccia and Troncy, 2005a] allows to find the best mappings (together with their weights) between the entities defined in the ontologies, using the prediction of several classifiers. These classifiers are terminological or machine learning-based, and we introduce a new one, that uses the semantics of the OWL axioms for establishing equivalence and subsumption relationships between the classes and the properties defined in the ontologies. *oMAP* can be downloaded for free [1].

Our approach is inspired by the data exchange problem [Fagin *et al.*, 2003] and borrows from others, like GLUE [Doan *et al.*, 2003a], the idea of using several specialized components for finding the best set of mappings. Theoretically, an ontology mapping in *oMAP* is a tuple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are respectively the source and target ontologies, and Σ is a finite set of *mapping constraints* of the form:

$$\alpha_{i,j}\, T_j \quad \leftarrow \quad S_i$$

[1] `http://homepages.cwi.nl/~troncy/oMAP/`

where S_i and T_j are respectively the source and target entities. The intended meaning of this rule is that the entity S_i of the source ontology is mapped onto the entity T_j of the target ontology, and the confident measure associated with this mapping is $\alpha_{i,j}$. Note that a source entity may be mapped onto several target entities and conversely. But, we do not require that we have a mapping for every target entity.

Aligning two ontologies in *oMap* consists of three steps:

1. We form a possible Σ, and estimate its quality based on the quality measures for its mapping rules;
2. For each mapping rule $T_j \leftarrow S_i$, we estimate its quality $\alpha_{i,j}$, which also depends on the Σ it belongs to, i.e. $\alpha_{i,j} = w(S_i, T_j, \Sigma)$;
3. As we cannot compute all possible Σ (there are exponentially many) and then choose the best one, we rather build iteratively our final set of mappings Σ using heuristics.

Similar to GLUE [Doan *et al.*, 2003a], we estimate the weight $w(S_i, T_j, \Sigma)$ of a mapping $T_j \leftarrow S_i$ by using different classifiers $CL_1, \ldots, CL_n$. Each classifier CL_k computes a weight $w(S_i, T_j, CL_k)$, which is the classifier's approximation of the rule $T_j \leftarrow S_i$. For each target entity T_j, CL_k provides a rank of the plausible source entities S_{i_k}. Then we rely on a priority list on the classifiers, $CL_1 \prec CL_2 \prec \ldots \prec CL_n$ and proceed as follows: for a given target entity T_j, select the top-ranked mapping of CL_1 if the weight is non-zero. Otherwise, select the top-ranked mapping provided by CL_2 if non-zero, and so on.

In the next section, we briefly present the classifiers that are currently used in our framework. It is worth noting that some of them consider the terminological part of the ontologies only, while others are based on their instances (i.e. the values of the individuals). Finally, we end this section by introducing a new classifier that fully uses the structure and the semantics of ontology definitions and axioms.

2.1 Terminological, Machine Learning-based and Structural Classifiers

The terminological classifiers work on the name of the entities (class or property) defined in the ontologies. In OWL, each resource is identified by a URI, and can have some annotation properties attached. Among others, the `rdfs:label` property may be used to provide a human-readable version of a resource's name. Furthermore, multilingual labels are supported using the language tagging facility of RDF literals. In the following, we consider that the name of an entity is given by the value of the `rdfs:label` property or by the URI fragment if this property is not specified. The typical terminological classifiers we used in *oMAP* compare the name of the entities, their stem (using the Porter stemming algorithm [Porter, 1980]), compute some similarity measures between the entity names (once downcased) such that the Levenshtein distance[Levenshtein, 1966] (or edit distance), or compute a similarity measure between the entity names using the WordNet®[2] relational dictionary.

[2] `http://wordnet.princeton.edu/`

Additionally, an ontology often contains some individuals. It is then possible to use machine learning-based classifiers to predict the weight of a mapping between two entities. The instances of an OWL ontology can be gathered using the following rules: we consider (i) the label for the named individuals, (ii) the data value for the datatype properties and (iii) the type for the anonymous individuals and the range of the object properties. For example, using the abstract syntax of [Horrocks *et al.*, 2003], let us consider the following individuals :

Individual (x_1 type `(Workshop)`
  value `(label "Italian Semantic Web Workshop")`
  value `(location` x_2`))`
Individual (x_2 type `(Address)`
  value `(city "Trento")` value `(country "Italy"))`

Then, the text gathered u_1 for the named individual x_1 will be (`"Italian Semantic Web Workshop", "Address"`) and u_2 for the anonymous individual x_2 (`"Address", "Trento", "Italy"`). Typical and well-known classifiers used in machine learning such as Naive Bayes and kNN [Sebastiani, 2002] have then been implemented in *oMAP* using these data.

Finally, we have drawn a new classifier which is able to use the semantics of the OWL definitions while being guided by their syntax. This *structural classifier* is fully described in [Straccia and Troncy, 2005b, Straccia and Troncy, 2005a]. It is used in the framework *a posteriori*. Indeed, we rely on the classifier preference relation $CL_{Name} \prec CL_{Stem} \prec CL_{EditDistance} \prec CL_{NaiveBayes}$. According to this preference relation, a set Σ' of mappings is determined. This set is given as input to the structural classifier. Then the structural classifier tries out all alternative ways to extend Σ' by adding some $T_j \leftarrow S_i$ if no mapping related to T_j is present in Σ'.

All the classifiers detailed previously have been implemented to be compatible with the alignment API [Euzenat, 2004], thus easing their chaining. Therefore, our *oMAP* framework benefits from all the evaluation facilities for comparing our approach with other methods. The problem of aligning ontologies has indeed already produced some interesting works. However, it is difficult to compare theoretically the various approaches proposed since they base on different techniques. Hence, it is necessary to compare them on common tests. This is the goal of the Ontology Alignment Evaluation Initiative (OAEI[3]) who set up evaluation campaign and benchmark tests for assessing the strengths and weakness of the available tools. We have evaluated *oMAP* with the data of the EON 2004 contest [Sure *et al.*, 2004] and we have participated actively to the 2005 campaign [Straccia and Troncy, 2005c].

2.2 Conclusion

As the number of Semantic Web applications is growing rapidly, many individual ontologies are created. The development of automated tools for ontology alignment will be of crucial importance. We have designed *oMAP*, a formal framework for ontology alignment, to cope this problem. *oMAP* uses different classifiers to estimate the quality of a mapping. Novel is the classifier which uses the structure of the OWL constructs and thus the semantics of the entities

[3] `http://oaei.inrialpes.fr`

defined in the ontologies. Furthermore, machine learning-based classifiers are employed. We have implemented the whole framework and evaluated it on independent benchmark tests provided by the Ontology Alignment Evaluation Initiative campaign.

As future work, we see some appealing points. Additional classifiers using more terminological resources can be included in the framework, and are currently under implementation while the effectiveness of the machine learning part could be improved using other measures like the kNN classifier or the KL-distance. While to fit new classifiers into our model is straightforward theoretically, practically finding out the most appropriate one or a combination of them is quite more difficult. In the future, more variants should be developed and evaluated to improve the overall quality of *oMAP*.

Chapter 3

Aligning Ontologies with Falcon

3.1 Overview

As an infrastructure for semantic web applications, Falcon[1] is a vision of our research group. It will provide enabling technologies for **F**inding, **A**ligning and **L**earning ontologies, and ultimately for **C**apturing knowledge by an **ON**tology-driven approach. It is still under development in our group. As a component of Falcon, Falcon-AO is an automatic tool for aligning ontologies. It is dedicated to aligning web ontologies expressed in OWL DL.

The overview of the system architecture of Falcon-AO is depicted in Fig.1. There are two matchers integrated in the current version (version 0.4): one is a matcher based on linguistic matching for ontologies, called LMO; and the other one is a matcher based on graph matching for ontologies, called GMO. The integration of the alignments generated by the two matchers is determined by the linguistic and structural comparability.

The main aligning process is outlined as follows:

1. Input two ontologies and parse them.
2. Observe the linguistic and structural comparability. In the case that both comparability are very low, the two ontologies are considered as totally different and Falcon-AO exits with no alignment.
3. Run LMO and obtain some alignments.
4. Set external entities of the ontologies according to the existing mapping pre-assigned by the system and the alignments generated by LMO.
5. Run GMO and obtain some additional alignments.
6. Integrate the alignments generated by LMO and GMO according to the linguistic and structural comparability.
7. Output the final alignments and exit.

[1] `http://xobjects.seu.edu.cn/project/falcon/falcon.htm`

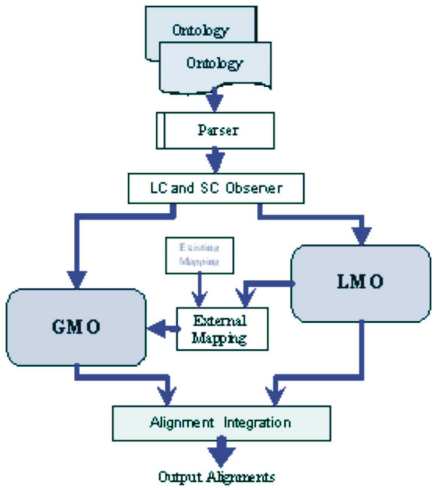

Figure 3.1: System Architecture

3.2 Specific Techniques

Three novel techniques are used in Falcon-AO. A brief introduction of these techniques are given in the following. More details can be found in [Hu *et al.*, 2005, Jian *et al.*, 2005, Qu *et al.*, 2005].

3.2.1 Linguistic Matching for Ontologies

LMO is based on an idea of virtual documents to pursue a cost-effective approach for linguistic matching. Basically, as a bag of weighted words, the virtual document of a URIref declared in an ontology contains not only the local descriptions but also the neighboring information to reflect the intended meaning of the URIref. Document similarity can be computed by traditional Vector Space techniques, and then be used in the similarity-based approaches to ontology matching.

3.2.2 Graph Matching for Ontologies

GMO uses bipartite graphs to represent ontologies, and measures the structural similarity between graphs. The idea of GMO is as follows: (a) similarity of two entities from two ontologies comes from the accumulation of similarities of involved statements (triples) taking the two entities as the same role (subject, predicate, object) in the triples; (b) the similarity of two statements comes from the accumulation of similarities of involved entities (including external entities) of the same role in the two statements being compared.

3.2.3 Linguistic vs. Structural Comparability

Falcon-AO integrates the matched entity pairs, which are generated by LMO and GMO, by observing the linguistic and structural comparability. The integration rules are described in brief as follows:

1. We take that linguistic comparability is somewhat more reliable than structural comparability, and that the alignments generated by LMO are always accepted by Falcon-AO.

2. When the linguistic comparability is high and the structural comparability is low, only alignments generated by GMO with high similarity are reliable and accepted by Falcon-AO.

3. If the linguistic comparability is low, all of the alignments generated by GMO are accepted by Falcon-AO. In this case, there is not enough information to measure these alignments and we can only assume that they are reliable.

3.3 Summary and Outlook

Falcon-AO is an automatic tool for aligning ontologies. Now, it integrates two matchers: LMO (A Linguistic Matching for Ontologies) and GMO (A Graph Matching for Ontologies). The experimental results on OAEI 2005 campaign demonstrate that Falcon-AO (version 0.3) performs very well on both Benchmark Test and Directory Test.

Some improvements will be considered in the future work: (a) the measurements of the linguistic and structural comparability of ontologies are still simple and an improvement will be needed, (b) the incorporation of corpus-based distributional similarity among words will be considered; and (c) some machine learning techniques will be integrated to realize a more powerful ontology matching tool.

Chapter 4

Ontology and Multimedia Ontology Alignment with ALIMO

In the effort to add multimedia documents in the Semantic Web multimedia ontologies will play an important role. In contrast to the usual ontologies, multimedia ontologies are formed by three different parts. The first part is the usual ontological part found in all web ontologies, which includes class, property and restriction definitions. The second part is the visual description part, where multimedia documents are given a visual description based on an MPEG-7 visual ontology. At last the third part is the actual raw data of the multimedia document. As it is obvious multimedia ontologies introduce new issues in task of (multimedia) ontology alignment that need to be tackled. For that purpose we are developing the platform ALIMO (Alignment of Multimedia Ontologies) which deals with all the features of multimedia ontologies.

The ALIMO platform consists of two matching modules. The first module is an ontology alignment method, which uses classical techniques for ontology alignment as the ones described in [Euzenat *et al.*, 2004]. The second module consists of a visual matching algorithm.

4.1 Ontology Alignment Module

The ALIMO platform uses three types of matching methods. These are the following:

- **Terminological Matching:** This method computes the similarities based on the strings of class and property names.
- **Structural Internal Matching:** In this method we refine the similarity computed by terminological matching, for two classes, by a portion of the similarities between the names of their properties.
- **Structural External Matching:** In this method we refine the similarity between two classes by a portion of the similarity computed for the super-classes of two classes.

For the assessment of the similarity between two class or property names ALIMO uses a novel string matching algorithm, called I-Sub Matching. This algorithm [Stoilos *et al.*, 2005], is an extension of the well known Sub-String matching method towards several directions. First of all

we believe that the similarity between two entities should be a function of both their commonalities as well as their differences. From that observation we have the following equation:

$$Sim(s_1, s_2) = Comm(s_1, s_2) - Diff(s_1, s_2) + winkler(s_1, s_2) \tag{4.1}$$

where $Comm(s_1, s_2)$ stands for the commonality between s_1 and s_2, $Diff(s_1, s_2)$ for the difference and $winkler(s_1, s_2)$ for the improvement of the result using the method introduced by Winkler in [Winkler, 1999]. Now, as a function of commonality we have used and extended the Substring distance metric. In contrast to the usual implementation, which searches only for the biggest common substring between two strings, we continue to find further common substrings until we have identified them all. Then we scale the length of the common substrings according to the following formula:

$$Comm(s_1, s_2) = \frac{2 * \sum_i length(maxComSubString_i)}{length(s_1) + length(s_2)} \tag{4.2}$$

As for the difference function, this is based on the length of the unmatched strings that have resulted from the initial matching step. Moreover, we believe that difference should play a less important role on the computation of the overall similarity. Our choice was the Hamacher product [Hamacher *et al.*, 1978], which is a parametric triangular norm. This leads us to the following equation:

$$Diff(s_1, s_2) = \frac{uLen_{s_1} * uLen_{s_2}}{p + (1 - p) * (uLen_{s_1} + uLen_{s_2} - uLen_{s_1} * uLen_{s_2})} \tag{4.3}$$

where $p \in [0, \infty)$, and $uLen_{s_1}$, $uLen_{s_2}$ represent the length of the unmatched substring from the initial strings s_1 and s_2 scaled with the string length, respectively.

Many ontology alignment algorithms use threshold values by which they determine which pairs of entities are to be considered similar and which not after a run of the algorithm. Obviously, the choice of the threshold is very crucial since a bad selection could remove many correct pairs or identify dissimilar ones as semantically equivalent. As pointed in [Stoilos *et al.*, 2005], one of the important features of the I-Sub method is that it improves the *stability* (threshold tolerance) of ontology alignment methods, compared to other string matching methods that exist in the literature. In other words, variations of the threshold of a platform from the optimal value will not affect the performance of the alignment platform, as is the case with most of the string matching methods.

In Figure 4.1 we can see our experimentation with ontology alignment using several popular string matching methods found in literature. The figure shows an average Recall versus average Precision chart relative to nine different threshold values used, ranging from 0.1 to 0.9. As we can see, all string matching methods achieve the best combination of precision and recall after the third/fourth threshold value (0.3/0.4). In terms of recall this can be interpreted to the interval from 0.8 to 0.83. From that point we can observe that if we increase (decrease) the threshold by one or two units we face a high degradation of the recall (precession), gaining in precision (recall). On

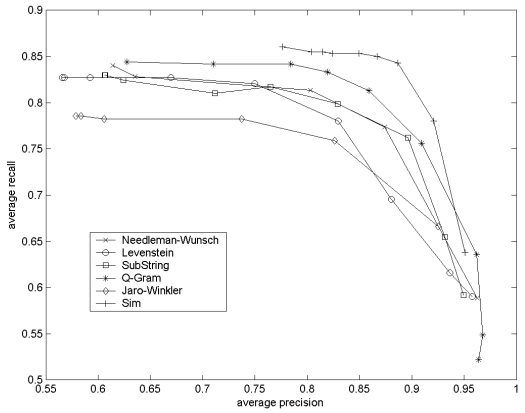

Figure 4.1: Average Precision vs. Average Recall values

the other hand the I-Sub method enjoys an area of 7 different threshold values, from 0.1 to 0.7, where precision can be increased, by increasing the threshold, while no or minor decrease in recall is encountered.

Chapter 5

FOAM – Framework for Ontology Alignment and Mapping

In recent years we have seen a range of research work on methods proposing alignments [Doan *et al.*, 2003b, Noy and Musen, 2003]. When one tries to apply these methods to some of the real-world scenarios of other research contributions [Ehrig *et al.*, 2003], one finds that existing alignment methods do not suit the given requirements: high quality results, efficiency, optional user-interaction, flexibility with respect to use cases, and easy adjustment and parametrization. The goal is to provide the end-user with a tool taking ontologies and returning alignments meeting these requirements. The Framework for Ontology Alignment and Mapping (FOAM[1]) itself consists of the general alignment process, specific extensions beyond its predecessor QOM, as presented in a previous deliverable, and pointers to the tool itself.

5.1 Alignment Process

One can observe that alignment methods like QOM [Ehrig and Sure, 2004] or PROMPT [Noy and Musen, 2003] may be mapped onto a generic alignment process (Figure 5.1). We refer to [Ehrig and Sure, 2004] for a detailed description. Here we will only mention the six major steps to clarify the underlying approach for the FOAM tool.

1. Feature Engineering, i.e. select excerpts of the overall ontology definition to describe a specific entity (e.g. label of an instance). FOAM makes use of all the features of OWL, including cardinality restrictions or enumeration definitions. Further domain-specific features may also be added.

2. Search Step Selection, i.e. choose two entities from the two ontologies to compare (e_1,e_2). Most approaches compare every entity of one ontology with every entity of the other ontology, but more efficient implementations are possible.

3. Similarity Assessment, i.e. indicate a similarity for a given description (feature) of two entities (e.g., $\text{sim}_{\text{superConcept}}(e_1,e_2)$=1.0).

[1] `http://www.aifb.uni-karlsruhe.de/WBS/meh/foam`

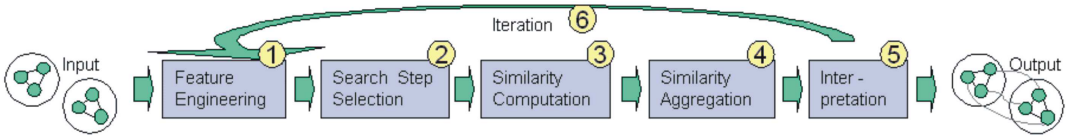

Figure 5.1: General Alignment Process

4. Similarity Aggregation, i.e. aggregate multiple similarity assessment for one pair of entities into a single measure.

5. Interpretation, i.e. use all aggregated numbers, a threshold and interpretation strategy to propose the alignment (align(e_1)='e_2').

6. Iteration, i.e. as the similarity of one entity pair influences the similarity of neighboring entity pairs, the equality is propagated through the ontologies.

Finally, we receive the alignments linking the two ontologies.

5.2 Extensions

Within the last year numerous additional methods extend the standard alignment process.

QOM – Quick Ontology Mapping: The QOM method [Ehrig and Sure, 2004] tackles the efficiency problem, which occurs when aligning larger ontologies. For this it makes use of the ontology structure. The number of candidate alignments to compare is considerably lowered by only allowing those which have very similar identifiers (or labels) or being a close neighbor of other existing alignments. Further, only those features are called on which do not require a complete traversing of the ontology, e.g., only direct instances of one concept are compared instead of all instances of all subconcepts. Both on theoretical and practical level the process is considerably sped up.

APFEL – Alignment Process Feature Estimation and Learning: Already the selection of which features to compare and which similarity measure to apply is very difficult. Setting aggregation weights for each feature is almost impossible, even for ontology experts. APFEL [Ehrig *et al.*, 2005] therefore is a method which solves these problems by using machine learning techniques. The user only has to provide some ontologies with known correct alignments. The learned decision tree is then used for aggregation and interpretation of the similarities.

Interactive Integration: So far the approaches focused on full-automation. However, it does make sense to include the user in the loop for some applications. By posing clever questions to the user he should be least bothered and at the same time receive best results. This is achieved by only presenting those candidate alignments to the user which are close to the threshold, i.e., for the system it is most uncertain whether they are true or false alignments. By manually tagging these accordingly quality of the results again increases considerably.

Adaptive Integration: The examination of several application scenarios [de Bruijn and Feier., 2005] has shown that the requirements for an alignment approach differ considerably, e.g., high efficiency versus high quality. FOAM has therefore been adapted once more. After the user has entered the scenario (alignment discovery, integration, merging, evolution, etc.) the parameters for the alignment process are chosen automatically [Ehrig and Sure, 2005]. This leads to better results, not in general, but for the specific scenario the alignments are required for. Thus, one implementation can be easily applied to several scenarios.

5.3 Implementation

The Framework for Ontology Alignment and Mapping (FOAM) has been implemented in Java. Further, it relies on the KAON2-environment[2] for processing ontologies (in specific ontologies represented in OWL-DL). This direct procedural approach can be very focused on specific problems arising for the alignment process, e.g., efficiency.

FOAM and its predecessors have been successfully applied in different applications. Within the SWAP-project,[3] FOAM was used to align and merge identical entities which were returned in the Bibster application or propose new aligned entities to the design board as needed in Xarop. Further, FOAM is a substantial part of the mediation component in the SEKT project.[4] Finally, the methods implemented in FOAM have been tested in three ontology alignment campaigns: I3CON, EON-OAC, and OAEI. FOAM behaved very favorable with results in the upper third of all systems, despite using only the standard full-automatic methods. Concrete results can be found in Part II.

FOAM is also an example of successful transition from research to industry. It has been integrated into the OntoMap tool, a graphical ontology mapping tool within the commercially sold OntoStudio framework of Ontoprise.[5]

The Framework for Ontology Alignment and Mapping is available through its webpage [6]. On the page one can find links to relevant publications, a download section of binaries and source code, installation guidelines and the documentation of FOAM, and some ontologies to test the tool. Further, there is a web-interface for internet users interested in very shallow testing. For real use is is recommend to download it.

[2] http://kaon2.semanticweb.org/
[3] http://swap.semanticweb.org/
[4] http://www.sekt-project.org/
[5] http://www.ontoprise.de/content/e3/e43/index_eng.html
[6] http://www.aifb.uni-karlsruhe.de/WBS/meh/foam

Chapter 6

OLA: OWL-Lite Alignment

OLA (for *O*WL-*L*ite *A*lignment) is an open-source tool jointly developed by teams at University of Montréal and INRIA Rhône Alpes. It features similarity-based alignment and a set of auxiliary services supporting the manipulation of alignment results [Euzenat and Valtchev, 2003, Euzenat and Valtchev, 2004].

Among the variety of alignment approaches (e.g., using machine learning, subsumption computation, formal concept analysis, etc.) similarity-based ones rely on a quantitative assessment of pair-wise likeness between entities. OLA, features a similarity model rooted in principles such as: completeness on the ontology language features, weighting of different feature contributions and mutual influence between related ontology entities. The resulting similarities are recursively defined hence their values are calculated by a step-wise, fixed-point-bound approximation process.

For the OAEI 2005 campaign, OLA was provided with an additional mechanism for weight determination that increased the autonomy of the system.

6.1 Overview

The primary goal behind the OLA tool design is to perform alignment of ontologies expressed in OWL, with a short-term emphasis on OWL-Lite and long-term one on OWL-DL. However, its GUI component, VISON[1] allows for many other services involving alignments (in the sense of [Euzenat, 2004]) to be accessed.

6.1.1 Functional specifications

From a mere algorithm for automated alignment construction, OLA has grown for the last year to an environment for alignment manipulation. Indeed, in its current version, the system offers, via its GUI component VISON, the following services:

- parsing and visualization of OWL-Lite and OWL-DL ontologies,
- computation of similarities between entities from two ontologies,
- extraction of alignments from a pair of ontologies, provided with a set of similarity matrices, one per category of ontology entities (see below),
- manual construction of alignments by composing entity pairs from two ontologies,

[1] http://www.iro.umontreal.ca/~owlola/

- use of an existing (partial) alignment as a seed for automated alignment construction (alignment completion),
- alignment visualization,
- comparison of two alignments.

In the remainder, the focus will be limited to the automated alignment construction with OLA.

6.1.2 Principles of matching in OLA

The following fundamental principles underly the design of the three key mechanisms in OLA – internal representation of the ontology, similarity computation and alignment extraction – that are involved in the global ontology alignment process:

All-encompassing comparison : We tend to believe that all the available knowledge about a pair of ontology entities should be taken into account when aligning. This does not exclude the possibility of ignoring particular aspects, i.g., OWL instances in case of OWL class comparison. However such a choice should be deliberately made by the tool user, here through appropriate weight assignment, or, if performed by an automated mechanisms, should reflect some particularity, either of the entire ontology (e.g., global absence of instances in both ontologies) or of the pair of entities at hand (e.g., local absence of instances in the pair of classes to be compared).

Highest automation level : Although we recognize that the entire alignment process often needs to be set on a semi-automated basis, we nevertheless argue in favor of a completely automated process for "draft" alignment generation. Thus, we see the OLA user providing a minimal set of parameters at the initial steps of the process whereas the tool will suggest one or more candidate alignments at the end, without any other human intervention.

Category-dependent comparison : Following the syntactic structure of the OWL language, entities are divided into categories, e.g., *classes*, *objects*, *properties*, *relations*, and only entities of the same category are compared. Moreover, the entities of a category are compared using similarity functions of the same basic shape. The respective category functions comprise the same factors and the same weights. They are further customized for each pair of category entities by projecting them over the actual feature space of the entities (which may be far smaller than the complete space of the category).

Comparability of similarity results : To enable comparison of similarity scores between different alignment tasks but also for some computational reasons, a set of useful properties is insured for the similarity functions: *normalization*, *positiveness*, *maximalness*[2], and *symmetry*[3].

6.1.3 Current limitations

- Although it would be valuable for alignment, OLA currently offers no inference mechanisms that could help complete the entity descriptions. In particular, inheritance is not used to expand entities, mostly out of efficiency considerations.
- Although neighborhoods play crucial role in the similarity definition, two neighbor entities are not necessarily affecting each other's respective similarities to a pair of other entities.

[2]With normalization, this amounts to forcing scores of 1 for identical entities within identical ontologies

[3]The price to pay for symmetry is the impossibility of detecting subsumption by this purely numerical procedure.

As only descriptive knowledge is taken into account, given two such entities, say e_1 and e_2, for e_2 to appear in a similarity expression for e_1, it should be considered as part of the description of the latter. For instance, a data type is not seen as being described by a property whose range the datatype represents. Consequently, datatypes are compared in an ontology-independent manner.

- Category borders are not similarity-permeable: Only entities from the same category are compared for similarity and hence for alignment.

6.1.4 Specific techniques used

OLA features an alignment process that splits into three basic steps: constructing the intermediate representation of the compared ontologies as labeled graphs, computing the similarity of each pair of same-category entities from the respective ontology graphs, extracting an alignment from the similarity matrices for each category.

6.1.5 OL-Graph construction

OL-Graphs are graph structures that provide an easy-to-process inner representation of OWL ontologies. An OL-Graph is a labeled graph where vertices correspond to OWL entities and edges to inter-entity relationships. As described in [Euzenat and Valtchev, 2004], the set of different vertex categories is: class (C), object (O), relation (R), property (P), property instance (A), datatype (D), datavalue (V), property restriction labels (L). Furthermore, we distinguish between datatype relations (R_{dt}) and object relations (R_o), and between datatype properties (P_{dt}) and object ones (P_o).

The OL-Graph model allows the following relationships among entities to be expressed:

- *specialization* between classes or relations (denoted $\mathcal{S}$),
- *instanciation* (denoted $\mathcal{I}$) between objects and classes, property instances and properties, values and datatypes,
- *attribution* (denoted $\mathcal{A}$) between classes and properties, objects and property instances;
- *restriction* (denoted $\mathcal{R}$) expressing the restriction on a property in a class,
- *valuation* (denoted $\mathcal{U}$) of a property in an object.

The OL-Graph of an ontology is built after the ontology is parsed[4]. The process of OL-Graph construction is described in [Tounazi, 2004].

6.1.6 Similarity model

The similarity functions used in OLA are designed in a category-specific manner and cover all the available descriptive knowledge about an entity pair. Thus, given a category X of OL-Graph nodes, the similarity of two nodes from X depends on:

- the similarities of the terms used to designate them, i.e., URIs, labels, names, etc.,
- the similarity of the pairs of neighbor nodes in the respective OL-Graphs that are linked by edges expressing the same relationships (e.g., class node similarity depends on similarity of superclasses, of property restrictions and of member objects),

[4] So far, we use the OWL API [Bechhofer *et al.*, 2003].

- the similarity of other local descriptive features depending on the specific category (e.g., cardinality intervals, property types)

Datatype and datavalue similarities are external to our model and therefore they are either user-provided or measured by a standard function (e.g., string identity of values and datatype names/URIs).

Formally, given a category X together with the set of relationships it is involved in, $\mathcal{N}(X)$, the similarity measure $Sim_X : X^2 \rightarrow [0, 1]$ is defined as follows:

$$Sim_X(x, x') = \sum_{\mathcal{F} \in \mathcal{N}(X)} \pi^X_{\mathcal{F}} MSim_Y(\mathcal{F}(x), \mathcal{F}(x')).$$

The function is normalized, i.e., the weights $\pi^X_{\mathcal{F}}$ sum to a unit, $\sum_{\mathcal{F} \in \mathcal{N}(X)} \pi^X_{\mathcal{F}} = 1$. for the computability The set functions $MSim_Y$ compare two sets of nodes of the same category (see [Euzenat and Valtchev, 2004] for details). Table 6.1 illustrates the set of similarities in our model.

OLA relies on various functions for identifiers comparison. Both string distances and lexical distances are used. Lexical distances rely on an exploration of WordNet 2.0 [Miller, 1995] with a quantitative assessment of the "relatedness" between two, possibly multi-word, terms. More specifically, the degree of relatedness between two WordNet entries is computed as the ratio between the depth, in graph-theoretic sense, of the most specific common hypernym and the average of both term depths. The computation of multi-word term similarity consists in first splitting the terms into a set of tokens each and then comparing all possible pairs of tokens from opposite sets using the above depth-based principle. The global term similarity is then computed as a similarity-based matching between both sets (see above).

As circular dependencies are impossible to avoid with the above definitions, the computation of the similarity values requires non-standard mechanisms. Following [Bisson, 1992, Valtchev, 1999], an equation system is composed out of the similarity definitions where variables correspond to similarities of node pairs while coefficients come from weights. The process of iterative, fixed-point-bound resolution of that system, as well as the related convergence and determinism issues are described in [Euzenat and Valtchev, 2004].

6.1.7 Implementation

OLA is implemented in JAVA. It is an implementation of the Alignment API [Euzenat, 2004] extending the standard implementation. OLA relies on the OWL API [Bechhofer *et al.*, 2003] for parsing OWL files. An entire subsystem is dedicated to the onstruction of OL-Graphs on top of the parsed ontologies. A set of further components that offer similarity computation services: substring distances, edit distances, Hamming distance, WordNet interface (via the JWNL library [Didion, 2004]), etc., that were originally designed for OLA are now part of the Alignment API. The VISON GUI component offers a uniform interface to all services provided by Alignment API and OLA. In particular, it visualizes both the input data, i.e., the OL-Graphs, and the final result, i.e., the alignment file, of the global process.

6.2 Improvements made for the 2004 evaluation

Several changes have been made to fit the complexity of the comparison. The most noteworthy one is the abandon of the requirement that all entities of the same category are compared along the

Funct.	Node	Factor	Measure
Sim_O	$o \in O$	$\lambda(o)$	sim_L
		$a \in A, (o,a) \in \mathcal{A}$	$MSim_A$
Sim_A	$a \in A$	$r \in R, (a,r) \in \mathcal{R}$	Sim_R
		$o \in O, (a,o) \in \mathcal{U}$	$MSim_O$
		$v \in V, (a,v) \in \mathcal{U}$	$MSim_V$
Sim_V	$v \in V$	value literal	type dependent
Sim_C	$c \in C$	$\lambda(c)$	sim_L
		$p \in P, (c,p) \in \mathcal{A}$	$MSim_P$
		$c' \in C, (c,c') \in \mathcal{S}$	$MSim_C$
sim_D	$d \in D$	$\lambda(r)$	XML-Schema
Sim_R	$r \in R$	$\lambda(r)$	sim_L
		$c \in C, (r, \texttt{domain}, c) \in \mathcal{R}$	$MSim_C$
		$c \in C, (r, \texttt{range}, c) \in \mathcal{R}$	$MSim_C$
		$d \in D, (r, \texttt{range}, d) \in \mathcal{R}$	Sim_D
		$r' \in R, (r,r') \in \mathcal{S}$	$MSim_R$
Sim_P	$p \in P$	$r \in R, (p,r') \in \mathcal{S}$	Sim_R
		$c \in C, (p, \texttt{all}, c) \in \mathcal{R}$	$MSim_C$
		$n \in \{0,1,\infty\}, (p, \texttt{card}, n) \in \mathcal{R}$	equality

Table 6.1: Similarity function decomposition (`card = cardinality` and `all = allValuesFrom`).

same feature space.

6.2.1 Adaptive description space

Following the lessons learned with our participation in the EON 2004 alignment contest [Euzenat and Valtchev, 2004], we found that the "uniform factor weights" condition tends to favor pairs of entities that have complete descriptions, i.e., pairs where both the members are connected to at least one descriptive entity for each of the similarity factors in the respective formula. Conversely, pairs where a particular factor is void tend to score to lesser similarity values. The extreme case is the pair of `Thing` classes which, if present, usually have almost no description. With fixed weights for similarity factors, and hence universal feature space for comparison, the `Thing` class pair will be evaluated to a relatively weak similarity value and the chances are high for it to be skipped from the alignment.

Consequently, we have adapted the above measure to fit cases where particular pair of entities is described only by a small subset of the entire set of category descriptors. Thus, a descriptive factor is ignored for similarity computation whenever neither of the compared entities possesses a neighbor with the underlying link label (e.g., no instances for a pair of compared classes). In this case, not only its weight is set to 0, but also the weights of the remaining "active" factors are increased correspondingly. To scale that principle up to the entire set of descriptive factors, the following simple mechanism has been realized in OLA: In order to keep both normalization and equity in similarity values, the weights of all non-null factors for a given entity pair are divided through their sum.

Thus, for a category X, the similarity measure $Sim^+_X : X^2 \rightarrow [0, 1]$ becomes:

$$Sim^+_X(x, x') = \frac{Sim_X(x, x')}{\sum_{\mathcal{F} \in \mathcal{N}^+(x,x')} \pi_{\mathcal{F}}}$$

where $\mathcal{N}^+(x, x')$ is the set of all relationships $\mathcal{F}$ for which $\mathcal{F}(x) \cup \mathcal{F}(x') \neq \emptyset$ [5].

6.2.2 Lexical similarity measure

The initial straightforward similarity measure has been replaced by a more sophisticated one that better accounts for semantic proximity between compound identifiers. Thus, given a pair of identifiers, they are first "tokenized", i.e., split into a set of atomic terms. Then, the respective pairs of terms are compared using WordNet. In fact, their degree of relatedness is computed as the ratio between the depth of the most specific common hypernym and the sum of both term depths. Finally, a similarity-based match is performed to establish a degree of proximity between the sets of terms.

6.2.3 Weight finding mechanism

To increase the level of automation in OLA, a weight-search mechanism was added to the initial architecture. Indeed, it is far from obvious for a novice user how to weight the different similarity factors. The underlying module performs several runs of the alignment producing subsystem with various weight combinations. It keeps only the combination that has resulted in the best alignment,

[5] That is, there exists at least one y such that $(x, y) \in \mathcal{F}$ or at least one y' such that $(x', y') \in \mathcal{F}$.

i.e., the one of the highest total similarity between aligned entities. On the one hand, this procedure is not realistic in a setting where reference alignments are not given. On the other hand, if the tests a realistic, then what is learned is the best behaviour of the system in general.

6.3 Improvements made for the 2005 evaluation

Along the preparation of the OAEI 2005 campaign, a row of changes have been made to the system in order to make it fit the complexity of the alignment discovery task. The most striking one is the introduction of a weight-computing mechanism that eliminates the necessity for the tool user to provide initial weights and hence makes a significant step towards full automation of the alignment process.

6.3.1 Weight computing mechanism

As it is far from obvious for novice users how to weigh the different similarity factors, we initiated work on incorporating a weight computing mechanism within the system. The intended mechanism is both intuitive and effective so that alignment practitioners with various skill levels could find a match for their knowledge and experience. So far, we used a simple heuristic method that, according to the obtained results, performs reasonably well. The basic idea of the method consists in distributing the weights among similarity factors in the generic similarity function of a node category according to the relative importance of the corresponding category in the entire ontology. That is to say we use the average number of links of the corresponding type per entity of the category at hand. For instance, the greater the number of super-class links in the ontology, the higher the weight of the super-class factor in the class similarity formula.

6.3.2 Similarity measure for entity names

OLA uses two alternative modes of comparison for entity names (URIs, labels, etc.): a string measure[6] (a default) and a lexical similarity measure that relies on WordNet 2.0 (see above).

The highly sophisticated lexical similarity measure that was used in OLA for the EON competition has been replaced by a simpler but more purposeful one. Indeed, the initial function compared multi-word terms on three separate axes: nouns, verbs and adjectives, as provided by WordNet 2,0. Such comparison seemed appropriate for cases where the meanings of a word fall in more than one part-of-speech category. The inter-word similarities on each axis were aggregated by an independent best-match computations while the three resulting values were further combined to a single one via a weighted sum.

The new measure trades separate matchings on speech-part-wise basis to a single global matching along entry similarities that aggregate all three possible aspects of a word. Thus, the words are compared to each other with all possible meanings and the highest similarity over a single pair of meanings is taken for the words.

For the OAEI competition, as we had to rely on a fixed parameter set for the entire collection of tests, we have chosen to force the use of the string distance. Indeed, it showed better performances while being much more efficient than the WordNet-based computation.

[6] subString distance provided by the Alignment API

Nevertheless, the improved lexical similarity was not completely discarded: it is currently used as a pre-processing tool that helps decide automatically the distribution of weights among similarity factors.

6.3.3 Minor adaptations

Following experiences from EON 2004, a set of simple but decisive modifications have been applied in order to prevent the precision leak in the tests. First, the instances have been excluded from the alignments by default, although the possibility is given to the user to reverse this choice. Then, entities external to the ontologies at hand have also been excluded from the alignment (but not from the similarity computation). Finally, one-to-one alignment production has been enforced in OLA to increase the potential recall of the resulting alignment.

6.4 Results

The results obtained in the OAEI-2005 evaluation are grouped by test categories.

6.4.1 Tests 1XX

OLA performed very well on the tests of this group. This seems to be due to the fact that while the language varies along the individual tests of the group, the basic ontology entities involved in the similarity computation remain unchanged with respect to the reference ontology.

6.4.2 Tests 2XX

The performances of the algorithm seem to suggest that three sub-groups of tests can be distinguished. The first one comprises the tests 21X, 22X, 23X and 24X, with a small number of exceptions where the performance have been:

- Quite good: This is the case of tests 201, 202, with random class names. The random names were putting a strain on the ability of the algorithm to propagate similarity along the network of node pairs. Obviously, our technique needs some improvements on that point.
- Satisfactory: In the case of tests 248, 249, there is a combination of missing (or random) names with one other missing factor. For tests 248, 249, the missing factors are hierarchy (sub-class links) and instances, respectively. Both play important role in similarity computation of classes, whenever these are stripped of their names as is the case with these two ontologies. Hence the sharp drop in precision and recall with respect to the preceding tests.
- Weak: The notorious failure here have been the tests 205, 209, which are the only ones to use of synonymous names in the ontology entities (with respect to the intial ontology). As WordNet has been plugged-out of the similarity computation, these results are not surprising.

The second groups is made of the tests 25X. Here OLA performances varied substantially: from extremely poor (254) to satisfactory (252, 259).

The last five ontologies of the group, the 26X ones, have proven to represent a serious obstacle for OLA. The performances of the system here were poor to very poor.

6.4.3 Tests 3XX

The real-world ontologies of the group 30X made OLA perform in an unimpressive way. We believe that this is due to the fact that string similarity was systematically used as identifier comparison means. Indeed, tentative runs with WordNet as basis for name similarity yielded way more precise alignments on that group. Unfortunately, they also brought down the overall statistics from the entire test set such as mean precision and mean recall. Hence the choice of the WordNet-based lexical similarity for a default name comparison means has been temporarily dropped.

6.4.4 Directory tests

We are glad to won this test especially since it was blind. However, the low level of recall shows that there is room for improvement (note that OLA is rather targeting ontologies in expressive languages so this kind of tests is not its primary target). We did not analyse the causes of failure so far.

6.4.5 Anatomy tests

We have not been able to load the tests due to our OWL Parser.

6.5 Conclusions

6.5.1 General comments

In its latest version, OLA has proven a more robust tool for alignment than it was a year before. The results show a substantial progress has been made since the EON 2004 alignment contest. With respect to the performances of OLA at that forum, we made a big leap amounting to about 25% in both mean precision and mean recall.

Nevertheless, we see that a vast space for improvement lays ahead of our project. The weaknesses of the current similarity mechanisms can be summarized as follows. First, the tuning of the algorithm is still a rigid process. Indeed, while the weights can now be computed following a specific footprint of the ontology, a mechanism for the choice of a particular name similarity on the same basis has yet to be defined.

Second, although we take into account the biggest possible amount of knowledge about entities, there are sources of similarity that have been ignored so far, in particular entity comments.

6.5.2 Discussions on the way to improve the proposed system

Besides expanding the lexical processing to comments in entities and providing a flexible decision mechanism for the choice of the default name similarity, a possible improvement of the system will be the integration of a learning module for weight estimation. As for similarity, the biggest challenge here is to define the representation of the input data, i.e., the descriptors of the entries for the learning algorithm.

Another research track would be the definition of an optimal matching algorithm. In fact, the current procedures are sub-optimal in the sense that they only chose local optima for each aligned entity. Consequently, as strict 1:1 matchings are to be produced, a single bad choice could easily

generate a chain of wrong alignment decisions and thus negatively impact the performances of the tool.

Part II

The Ontology Alignment Challenge 2005

Chapter 7

OAEI-2005: organization

The increasing number of methods available for schema matching/ontology integration suggests the need to establish a consensus for evaluation of these methods. The Ontology Alignment Evaluation Initiative[1] is now a coordinated international initiative that has been set up for organizing evaluation of ontology matching algorithms. After the two events organized in 2004 (namely, the Information Interpretation and Integration Conference (I3CON) and the EON Ontology Alignment Contest [Sure *et al.*, 2004]), this year one unique evaluation campaign is organized. Its outcome is presented at the Workshop on Integrating Ontologies held in conjunction with K-CAP 2005 at Banff (Canada) on October 2, 2005. Since last year, we have set up a web site, improved the software on which the tests can be evaluated and set up some precise guidelines for running these tests. We have taken into account last year's remarks by (1) adding more coverage to the benchmark suite and (2) elaborating two real world test cases (as well as addressing other technical comments).

This chapter serves as a presentation to the 2005 evaluation campaign and introduction to the results provided by the some of the systems presented in the previous papers.

7.1 Goals

Last year events demonstrated that it is possible to evaluate ontology alignment tools. One intermediate goal of this year is to take into account the comments from last year contests. In particular, we aimed at improving the tests by widening their scope and variety. Benchmark tests are more complete (and harder) than before. Newly introduced tracks are more 'real-world' and of a considerable size. The main goal of the Ontology Alignment Evaluation is to be able to compare systems and algorithms on the same basis and to allow drawing conclusions about the best strategies. Our ambition is that from such challenges, the tool developers can learn and improve their systems.

7.2 General methodology

We present below the general methodology for the 2005 campaign. In this we took into account many of the comments made during the previous campaign.

[1] `http://oaei.inrialpes.fr`

7.2.1 Alignment problems

This year's campaign consists of three parts: it features two real world blind tests (anatomy and directory) in addition to the systematic benchmark test suite. By blind tests it is meant that the result expected from the test is not known in advance by the participants. The evaluation organizers provide the participants with the pairs of ontologies to align as well as (in the case of the systematic benchmark suite only) expected results. The ontologies are described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in a standard format expressed in RDF/XML [Euzenat, 2004].

- Like for last year's EON contest, a systematic benchmark series has been produced. The goal of this benchmark series is to identify the areas in which each alignment 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 directory real world case consists of aligning web sites directory (like open directory or Yahoo's). It is more than two thousand elementary tests.
- The anatomy real world case covers the domain of body anatomy and consists of two ontologies with an approximate size of several 10k classes and several dozen of relations.

The evaluation has been processed in three successive steps.

7.2.2 Preparatory phase

The ontologies and alignments of the evaluation have been provided in advance during the period between June 1st and July 1st. This was the occasion for potential participants to send observations, bug corrections, remarks and other test cases to the organizers. The goal of this primary period is to be sure that the delivered tests make sense to the participants. The feedback is important, so all participants were strongly invited to provide it. The final test base has been released on July 4th. The tests did only change after this period for ensuring a better and easier participation.

7.2.3 Execution phase

During the execution phase the participants have used their algorithms to automatically match the ontologies of both part. The participants were required to only use one algorithm and the same set of parameters for all tests. Of course, it is regular to select the set of parameters that provide the best results. Beside the parameters the input of the algorithms must be the two provided ontology to align and any general purpose resource available to everyone (that is no resource especially designed for the test). In particular, the participants should not use the data (ontologies and results) from other test sets to help their algorithm.

The participants have provided their alignment for each test in the Alignment format and a paper describing their results. In an attempt to validate independently the results, they were required to provide a link to their program and parameter set used for obtaining the results.

Name	System	Benchmarks	Directory	Anatomy	Validated	Relations	Confidence
U. Karlsruhe	FOAM	√	√			=	cont
U. Montréal/INRIA	OLA	√	√		√	=	cont
IRST Trento	CtxMatch 2	√	√			=, ≤	1/0
U. Southampton	CMS	√	√	√		=	1/0
Southeast U. Nanjin	Falcon	√	√	√	√	=	1/0
UC. Dublin	?	√	√			=	cont
CNR/Pisa	OMAP	√	√			=	1/0

Table 7.1: Participants and the state of the state of their submissions. Confidence is given as 1/0 or continuous values.

7.2.4 Evaluation phase

The organizers have evaluated the results of the algorithms used by the participants and provided comparisons on the basis of the provided alignments. In the case of the real world ontologies only the organizers did the evaluation with regard to the withheld alignments. The standard evaluation measures are precision and recall computed against the reference alignments. For the matter of aggregation of the measures we have computed a true global precision and recall (not a mere average). We have also computed precision/recall graphs for some of the participants (see below). Finally, in an experimental way, we have attempted this year at reproducing the results provided by participants (validation).

7.3 Comments on the execution

We had more participants than last year's event and it is easier to run these tests (qualitatively we had less comments and the results were easier to analyze). We summarize the list of participants in Table 7.1. As can be seen, not all participants provided results for all the tests and not all system were correctly validated. However, when the tests are straightforward to process (benchmarks and directory), participants provided results. The main problems with the anatomy test was its size. We also mentioned the kind of results sent by each participant (relations and confidence).

We note that the time devoted for performing these tests (three months) and the period allocated for that (summer) is relatively short and does not really allow the participants to analyze their results and improve their algorithms. On the one hand, this prevents having algorithms really tuned for the test set, on the other hand, this can be frustrating for the participants. We should try to allow more time for participating next time.

Complete results are provided on http://oaei.inrialpes.fr/2005/results/. These are the only official results (the results presented here are only partial and prone to correction). The summary of results track by track is provided below.

Chapter 8

OAEI-2005: results

8.1 Benchmark

The benchmark test case improved on last year's base by providing new variations of the reference ontology (last year the test contained 19 individual tests while this year it contains 53 tests). These new tests are supposed to be more difficult. The other improvement was the introduction of other evaluation metrics (real global precision and recall as well as the generation of precision-recall graphs).

8.1.1 Test set

The systematic benchmark test set is built around one reference ontology and many variations of it. The participants have to match this reference ontology with the variations. These variations are focussing the characterization of the behavior of the tools rather than having them compete on real-life problems. The ontologies are described in OWL-DL and serialized in the RDF/XML format. 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 last year EON Ontology Alignment Contest. Test numbering (almost) fully preserves the numbering of the first EON contest.

The reference ontology is based on the one of the first EON Ontology Alignment Contest. It is improved by comprising a number of circular relations that were missing from the first test. 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 (based on area, quality, etc.). We choose the one common among scholars based on mean of publications; as many ontologies below (tests #301-304), it is reminiscent to BibTeX. 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. The reference ontology is put in the context of the semantic web by using other external resources for expressing non bibliographic information. It takes advantage of FOAF[1] and iCalendar[2] for expressing the People, Organization and Event concepts. Here are the external reference used:

[1] `http://xmlns.com/foaf/0.1/`
[2] `http://www.w3.org/2002/12/cal/`

algo	edna		falcon		foam		ctxMatch2-1		dublin20		cms		omap		ola	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	0.96	1.00	1.00	1.00	0.98	0.65	0.10	0.34	1.00	0.99	0.74	0.20	0.96	1.00	1.00	1.00
2xx	0.41	0.56	0.90	0.89	0.89	0.69	0.08	0.23	0.94	0.71	0.81	0.18	0.31	0.68	0.80	0.73
3xx	0.47	0.82	0.93	0.83	0.92	0.69	0.08	0.22	0.67	0.60	0.93	0.18	0.93	0.65	0.50	0.48
H-means	0.45	0.61	0.91	0.89	0.90	0.69	0.08	0.24	0.92	0.72	0.81	0.18	0.35	0.70	0.80	0.74

Table 8.1: Means of results obtained by participants (corresponding to harmonic means)

- http://www.w3.org/2002/12/cal/#:Vevent (defined in http://www.w3.org/2002/12/cal/ical.n3 and supposedly in http://www.w3.org/2002/12/cal/ical.rdf)
- http://xmlns.com/foaf/0.1/#:Person (defined in http://xmlns.com/foaf/0.1/index.rdf)
- http://xmlns.com/foaf/0.1/#:Organization (defined in http://xmlns.com/foaf/0.1/index.rdf)

This reference ontology is a bit limited in the sense that it does not contain attachment to several classes. Similarly the kind of proposed alignments is still limited: they only match named classes and properties, they mostly use the "=" relation with confidence of 1. There are still three group of tests in this benchmark:

- simple tests (1xx) such as comparing the reference ontology with itself, with another irrelevant ontology (the wine ontology used in the OWL primer) or the same ontology in its restriction to OWL-Lite;
- systematic tests (2xx) that were obtained by discarding some features of the reference ontology. The considered features were (names, comments, hierarchy, instances, relations, restrictions, etc.). The tests are systematically generated to as to start from some reference ontology and discarding a number of information in order to evaluate how the algorithm behave when this information is lacking. These tests were largely improved from last year by combining all feature discarding.
- four real-life ontologies of bibliographic references (3xx) that were found on the web and left mostly untouched (they were added xmlns and xml:base attributes).

Table 8.4 summarize what has been retracted from the reference ontology in the systematic tests. There are here 6 categories of alteration:

Name Name of entities that can be replaced by (R/N) random strings, (S)ynonyms, (N)ame with different conventions, (F) strings in another language than english.
Comments Comments can be (N) suppressed or (F) translated in another language.
Specialization Hierarchy can be (N) suppressed, (E)xpansed or (F)lattened.
Instances can be (N) suppressed
Properties can be (N) suppressed or (R) having the restrictions on classes discarded.
Classes can be (E)xpanded, i.e., replaced by several classes or (F)latened.

8.1.2 Results

Table 8.1 provide the consolidated results, by groups of tests. Table 8.5 contain the full results.

We display the results of participants as well as those given by some very simple edit distance algorithm on labels (edna). The computed values here are real precision and recall and not a simple average of precision and recall. This is more accurate than what has been computed last

algo	karlsruhe2		umontreal		fujitsu		stanford	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	NaN	0.00	0.57	0.93	0.99	1.00	0.99	1.00
2xx	0.60	0.46	0.54	0.87	0.93	0.84	0.98	0.72
3xx	0.90	0.59	0.36	0.57	0.60	0.72	0.93	0.74
H-means	0.65	0.40	0.52	0.83	0.88	0.85	0.98	0.77

Table 8.2: EON 2004 results with this year's aggregation method.

algo	edna		falcon		foam		ctxMatch2-1		dublin20		cms		omap		ola	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	0.96	1.00	1.00	1.00	0.98	0.65	0.10	0.34	1.00	0.99	0.74	0.20	0.96	1.00	1.00	1.00
2xx	0.66	0.72	0.98	0.97	0.87	0.73	0.09	0.25	0.98	0.92	0.91	0.20	0.89	0.79	0.89	0.86
3xx	0.47	0.82	0.93	0.83	0.92	0.69	0.08	0.22	0.67	0.60	0.93	0.18	0.93	0.65	0.50	0.48
H-means	0.66	0.78	0.97	0.96	0.74	0.59	0.09	0.26	0.94	0.88	0.65	0.18	0.90	0.81	0.85	0.83

Table 8.3: This year's results on EON 2004 test bench.

year.

As can be seen, the 1xx tests are relatively easy for most of the participants. The 2xx tests are more difficult in general while 3xx tests are not significantly more difficult than 2xx for most participants. The real interesting results is that there are significant differences across algorithms within the 2xx test series. Most of the best algorithms were combining different ways of finding the correspondence. Each of them is able to perform quite well on some tests with some methods. So the key issue seems to have been the combination of different methods (as described by the papers).

One algorithm, Falcon, seems largely dominant. But a group of other algorithms (Dublin, OLA, FOAM) are competing against each other, while the CMS and CtxMatch currently perform at a lower rate. Concerning these algorithm, CMS seems to privilege precision and performs correctly in this (OLA seems to have privileged recall with regard to last year). CtxMatch has the difficulty of delivering many subsumption assertions. These assertions are taken by our evaluation procedure positively (even if equivalence assertions were required), but since there are many more assertions than in the reference alignments, this brings the result down.

These results can be compared with last year's results given in Table 8.2 (with aggregated measures computed at new with the methods of this year). For the sake of comparison, the results of this year on the same test set as last year are given in Table 8.3. As can be expected, the two participants of both challenges (Karlsruhe2 corresponding to foam and Montréal/INRIA corresponding to ola) have largely improved their results. The results of the best participants this year are over or similar to those of last year. This is remarkable, because participants did not tune their algorithms to the challenge of last year but to that of this year (more difficult since it contains more test of a more difficult nature and because of the addition of cycles in them).

So, it seem that the field is globally progressing.

Because of the precision/recall trade-off, as noted last year, it is difficult to compare the middle group of systems. In order to assess this, we attempted to draw precision recall graphs. We provide in Figure 8.1 the averaged precision and recall graphs of this year. They involve only the results of all participants. However, the results corresponding to participants who

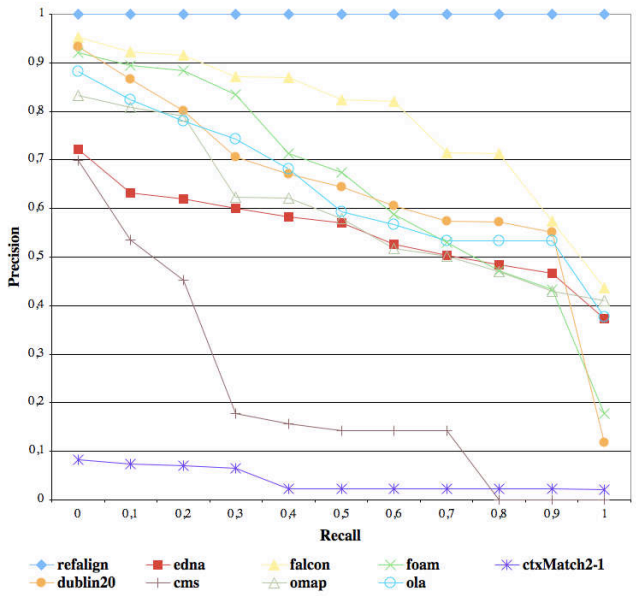

Figure 8.1: Precision-recall graphs

provided confidence measures equal to 1 or 0 (see Table 7.1) can be considered as approximation. Moreover, for reason of time these graphs have been computed by averaging the graphs of each tests (instead to pure precision and recall).

These graphs are not totally faithful to the algorithms because participants have cut their results (in order to get high overall precision and recall). However, they provide a rough idea about the way participants are fighting against each others in the precision recall space. It would be very useful that next year we ask for results with continuous ranking for drawing these kind of graphs.

8.1.3 Comments

As general comments, we remarks that it is still difficult for participants to provide results that correspond to the challenge (incorrect format, alignment with external entities). Because time is short and we try to avoid modifying provided results, this test is still a test of both algorithms and their ability to deliver a required format. However, some teams are really effective in this (and the same teams generally have their tools validated relatively easily).

The evaluation of algorithms like ctxMatch which provide many subsumption assertions is relatively inadequate. Even if the test can remain a test of inference equivalence. It would be useful to be able to count adequately, i.e., not negatively for precision, true assertions like owl:Thing subsuming another concept. We must develop new evaluation methods taken into account these assertions and the semantics of the OWL language.

8.2 Directory

8.2.1 Data set

The data set exploited in the web directories matching task was constructed from Google, Yahoo and Looksmart web directories as described in [Avesani *et al.*, 2005]. The key idea of the data set construction methodology was to significantly reduce the search space for human annotators. Instead of considering the full mapping task which is very big (Google and Yahoo directories have up to $3 * 10^5$ nodes each: this means that the human annotators need to consider up to $(3*10^5)^2 = 9*10^{10}$ mappings), it uses semi automatic pruning techniques in order to significantly reduce the search space. For example, for the dataset described in [Avesani *et al.*, 2005] human annotators consider only 2265 mappings instead of the full mapping problem.

The major limitation of the current dataset version is the fact that if it contains true positive mappings (i.e., it is correct), it does not contain them all (it is not complete). Notice that manually constructed mapping sets (such as ones presented for systematic tests) assume all the mappings except true positives to be true negatives (i.e., they are supposed to be complete). This limitation allows to use the dataset only for evaluation of Recall (since Recall is defined as ratio of correct mappings found by the system to the total number of correct mappings, this ratio is still meaningful if we only know a part of the correct mappings). At the same time measuring Precision necessarily requires the completeness in the dataset since Precision is defined as a ratio of correct mappings found by the system to all the mappings found by the system: in this case if we only know one part of the correct mapping it is possible that a better performing system have a worse precision on the test set.

The absence of completeness has significant implications on the testing methodology in general. In fact most of the state of the art matching systems can be tuned either to produce the results with better Recall or to produce the results with better Precision. For example, the system which produce the complete product relation on any input will always have 100% Recall. Therefore, the main methodological goal in the evaluation was to prevent Recall tuned systems from getting of unrealistically good results on the dataset. In order to accomplish this goal the double validation of the results was performed. The participants were asked for the binaries of their systems and were required to use the same sets of parameters in both web directory and systematic matching tasks. Then the results were double checked by organizers to ensure that the latter requirement is fulfilled by the authors. The process allow to recognize Recall tuned systems by analysis of systematic tests results.

The dataset originally was presented in its own format. The mappings were presented as pairwise relationships between the nodes of the web directories identified by their paths to root. Since the systems participating in the evaluation all take OWL ontologies as input the conversion of the dataset to OWL was performed. In the conversion process the nodes of the web directories were modeled as classes and classification relation connecting the nodes was modeled as rdfs:subClassOf relation. Moreover, in order to avoid presenting a too big challenge for matchers, the matching task was presented as 2265 tasks of finding the semantic relation holding between pathes to root in the web directories modeled as sub class hierarchies.

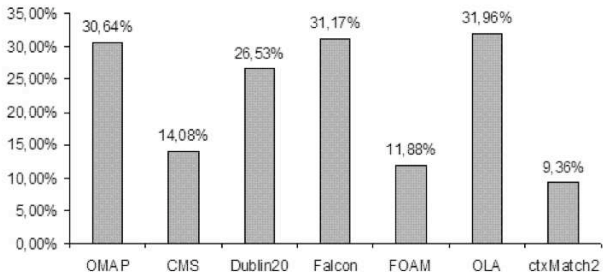

Figure 8.2: Recall for web directories matching task

8.2.2 Results

The results for web directory matching task are presented on Figure 8.2. As from the figure the web directories matching task is a very hard one. In fact the best systems found about 30% of mappings form the dataset (i.e., have Recall about 30%).

The evaluation results can be considered from two perspectives. On the one hand, they are good indicator of real world ontologies matching complexity. On the other hand, the results can provide information about the quality of the dataset used in the evaluation. The desired mapping dataset quality properties were defined in [Avesani *et al.*, 2005] as *Complexity*, *Discrimination capability*, *Incrementality* and *Correctness*. The first means that the dataset is "hard" for state of the art matching systems, the second that it discriminates among the various matching solutions, the third that it is effective in recognizing weaknesses in the state of the art matching systems and the fourth that it can be considered as a correct one.

The results of the evaluation give us some evidence for *Complexity* and *Discrimination capability* properties. As from Figure 8.2 TaxME dataset is hard for state of the art matching techniques since there are no systems having Recall more than 35% on the dataset. At the same time all the matching systems together found about 60% of mappings. This means that there is a big space for improvements for state of the art matching solutions.

Consider Figure 8.3. It contains partitioning of the mappings found by the matching systems. As from the figure 44% of the mappings found by any of the matching systems was found by only one system. This is a good argument to the dataset *Discrimination capability* property.

8.2.3 Comments

The web directories matching task is an important step towards evaluation on the real world matching problems. At the same time there are a number of limitations which makes the task only an intermediate step. First of all the current version of the mapping dataset provides correct but not complete set of reference mappings. The new mapping dataset construction techniques can overcome this limitation. In the evaluation the mapping task was split to the the tiny subtasks. This strategy allowed to obtain results form all the matching systems participating in the evaluation.

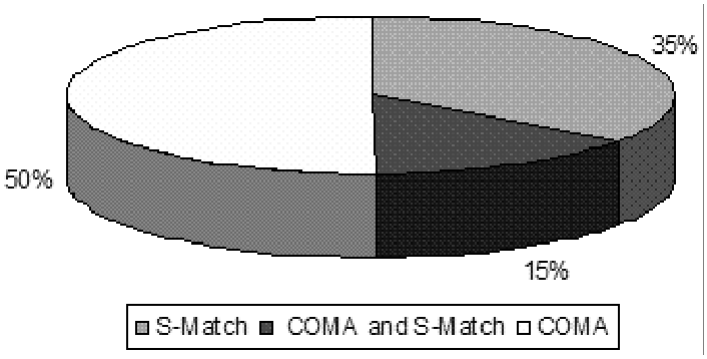

Figure 8.3: Partitioning of the mappings found by the matching systems

At the same time it hides computational complexity of "real world" matching (the web directories have up to 10^5 nodes) and may affect the results of the tools relying on "look for similar siblings" heuristic.

The results obtained on the web directories matching task coincide well with previously reported results on the same dataset. According to [Avesani *et al.*, 2005] generic matching systems (or the systems intended to match any graph-like structures) have Recall from 30% to 60% on the dataset. At the same time the real world matching tasks are very hard for state of the art matching systems and there is a huge space for improvements in the ontology matching techniques.

8.3 Anatomy

8.3.1 Test set

The focus of this task is to confront existing alignment technology with real world ontologies. Our aim is to get a better impression of where we stand with respect to really hard challenges that normally require an enormous manual effort and requires in-depth knowledge of the domain. The task is placed in the medical domain as this is the domain where we find large, carefully designed ontologies. The specific characteristics of the ontologies are:

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

This implies that the task will be challenging from a technological point of view, but there is guidance for tuning matching approach that needs to be taken into account.

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

The Foundational Model of Anatomy

The Foundational Model of Anatomy is a medical ontology developed by the University of Washington. We extracted an OWL version of the ontology from a Protege database. The model contains the following information:

- Class hierarchy;
- Relations between classes;
- Free text documentation and definitions of classes;
- Synonyms and names in different languages.

The OpenGalen Anatomy Model

The second ontology is the Anatomy model developed in the OpenGalen Project by the University of Manchester. We created an OWL version of the ontology using the export functionality of Protege. The model contains the following information:

- Concept hierarchy;
- Relations between concepts.

The task is to find alignment between classes in the two ontologies. In order to find the alignment, any information in the two models can be used. In addition, it is allowed to use background knowledge, that has not specifically been created for the alignment tasks (i.e., no hand-made mappings between parts of the ontologies). Admissible background knowledge are other medical terminologies such as UMLS as well as medical dictionaries and document sets. Further, results must not be tuned manually, for instance, by removing obviously wrong mappings.

At the time of printing we are not able to provide results of evaluation on this test.

8.3.2 Comments

We had very few participants able to even produce the alignments between both ontologies. This is mainly due to their inability to load these ontologies with current OWL tools (caused either by the size of the ontologies or errors in the OWL).

8.4 Result validation

As can be seen from the procedure, the results are not obtained independently. They have been computed from the alignment provided by the participants. In order to go one step further, we have attempted, this year, to generate the results obtained by the participants from their tools. The tools for which the results have been validated independently are marked in Table 7.1.

#	Name	Com	Hier	Inst	Prop	Class	Comment
101							Reference alignment
102							Irrelevant ontology
103							Language generalization
104							Language restriction
201	R						No names
202	R	N					No names, no comments
203		N					No comments (was missspelling)
204	C						Naming conventions
205	S						Synonyms
206	F	F					Translation
207	F						
208	C	N					
209	S	N					
210	F	N					
221			N				No specialisation
222			F				Flatenned hierarchy
223			E				Expanded hierarchy
224				N			No instance
225					R		No restrictions
226							No datatypes
227							Unit difference
228					N		No properties
229							Class vs instances
230						F	Flattened classes
231*						E	Expanded classes
232			N	N			
233			N		N		
236				N	N		
237			F	N			
238			E	N			
239			F		N		
240			E		N		
241			N	N	N		
246			F	N	N		
247			E	N	N		
248	N	N	N				
249	N	N		N			
250	N	N			N		
251	N	N	F				
252	N	N	E				
253	N	N	N	N			
254	N	N	N		N		
257	N	N		N	N		
258	N	N	F	N			
259	N	N	E	N			
260	N	N	F		N		
261	N	N	E		N		
262	N	N	N	N	N		
265	N	N	F	N	N		
266	N	N	E	N	N		
301							Real: BibTeX/MIT
302							Real: BibTeX/UMBC
303							Real: Karlsruhe
304							Real: INRIA

Table 8.4: Structure of the systematic benchmark test-case

algo	edna		falcon		foam		ctxMatch2-1		dublin20		cms		omap		ola	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
101	0.96	1.00	1.00	1.00	n/a	n/a	0.10	0.34	1.00	0.99	n/a	n/a	0.96	1.00	1.00	1.00
103	0.96	1.00	1.00	1.00	0.98	0.98	0.10	0.34	1.00	0.99	0.67	0.25	0.96	1.00	1.00	1.00
104	0.96	1.00	1.00	1.00	0.98	0.98	0.10	0.34	1.00	0.99	0.80	0.34	0.96	1.00	1.00	1.00
201	0.03	0.03	0.98	0.98	n/a	n/a	0.00	0.00	0.96	0.96	1.00	0.07	0.80	0.38	0.71	0.62
202	0.03	0.03	0.87	0.87	0.79	0.52	0.00	0.00	0.75	0.28	0.25	0.01	0.82	0.24	0.66	0.56
203	0.96	1.00	1.00	1.00	1.00	1.00	0.08	0.34	1.00	0.99	1.00	0.24	0.96	1.00	1.00	1.00
204	0.90	0.94	1.00	1.00	1.00	0.97	0.09	0.28	0.98	0.98	1.00	0.24	0.93	0.89	0.94	0.94
205	0.34	0.35	0.88	0.87	0.89	0.73	0.05	0.11	0.98	0.97	1.00	0.09	0.58	0.66	0.43	0.42
206	0.51	0.54	1.00	0.99	1.00	0.82	0.05	0.08	0.96	0.95	1.00	0.09	0.74	0.49	0.94	0.93
207	0.51	0.54	1.00	0.99	0.96	0.78	0.05	0.08	0.96	0.95	1.00	0.09	0.74	0.49	0.95	0.94
208	0.90	0.94	1.00	1.00	0.96	0.89	0.09	0.28	0.99	0.96	1.00	0.19	0.96	0.90	0.94	0.94
209	0.35	0.36	0.86	0.86	0.78	0.58	0.05	0.11	0.68	0.56	1.00	0.04	0.41	0.60	0.43	0.42
210	0.51	0.54	0.97	0.96	0.87	0.64	0.05	0.08	0.96	0.82	0.82	0.09	0.88	0.39	0.95	0.94
221	0.96	1.00	1.00	1.00	1.00	1.00	0.12	0.34	1.00	0.99	1.00	0.27	0.96	1.00	1.00	1.00
222	0.91	0.99	1.00	1.00	0.98	0.98	0.11	0.31	1.00	0.99	1.00	0.23	0.96	1.00	1.00	1.00
223	0.96	1.00	1.00	1.00	0.99	0.98	0.09	0.34	0.99	0.98	0.96	0.26	0.96	1.00	1.00	1.00
224	0.96	1.00	1.00	1.00	1.00	0.99	0.10	0.34	1.00	0.99	1.00	0.27	0.96	1.00	1.00	1.00
225	0.96	1.00	1.00	1.00	0.00	0.00	0.08	0.34	1.00	0.99	0.74	0.26	0.96	1.00	1.00	1.00
228	0.38	1.00	1.00	1.00	1.00	1.00	0.12	1.00	1.00	1.00	0.74	0.76	0.92	1.00	1.00	1.00
230	0.71	1.00	0.94	1.00	0.94	1.00	0.08	0.35	0.95	0.99	1.00	0.26	0.89	1.00	0.95	0.97
231	0.96	1.00	1.00	1.00	0.98	0.98	0.10	0.34	1.00	0.99	1.00	0.27	0.96	1.00	1.00	1.00
232	0.96	1.00	1.00	1.00	1.00	0.99	0.12	0.34	1.00	0.99	1.00	0.27	0.96	1.00	1.00	1.00
233	0.38	1.00	1.00	1.00	1.00	1.00	0.12	1.00	1.00	1.00	0.81	0.76	0.92	1.00	1.00	1.00
236	0.38	1.00	1.00	1.00	1.00	1.00	0.09	1.00	1.00	1.00	0.74	0.76	0.92	1.00	1.00	1.00
237	0.91	0.99	1.00	1.00	1.00	0.99	0.11	0.31	1.00	0.99	1.00	0.23	0.95	1.00	0.97	0.98
238	0.96	1.00	0.99	0.99	1.00	0.99	0.07	0.34	0.99	0.98	0.96	0.26	0.96	1.00	0.99	0.99
239	0.28	1.00	0.97	1.00	0.97	1.00	0.14	1.00	0.97	1.00	0.71	0.76	0.85	1.00	0.97	1.00
240	0.33	1.00	0.97	1.00	0.94	0.97	0.10	1.00	0.94	0.97	0.71	0.73	0.87	1.00	0.97	1.00
241	0.38	1.00	1.00	1.00	1.00	1.00	0.12	1.00	1.00	1.00	0.81	0.76	0.92	1.00	1.00	1.00
246	0.28	1.00	0.97	1.00	0.97	1.00	0.14	1.00	0.97	1.00	0.71	0.76	0.85	1.00	0.97	1.00
247	0.33	1.00	0.94	0.97	0.94	0.97	0.10	1.00	0.94	0.97	0.71	0.73	0.87	1.00	0.97	1.00
248	0.06	0.06	0.84	0.82	0.89	0.51	0.00	0.00	0.71	0.25	0.25	0.01	0.82	0.24	0.59	0.46
249	0.04	0.04	0.86	0.86	0.80	0.51	0.00	0.00	0.74	0.29	0.25	0.01	0.81	0.23	0.59	0.46
250	0.01	0.03	0.77	0.70	1.00	0.55	0.00	0.00	1.00	0.09	0.00	0.00	0.05	0.45	0.30	0.24
251	0.01	0.01	0.69	0.69	0.90	0.41	0.00	0.00	0.79	0.32	0.25	0.01	0.82	0.25	0.42	0.30
252	0.01	0.01	0.67	0.67	0.67	0.35	0.00	0.00	0.57	0.22	0.25	0.01	0.82	0.24	0.59	0.52
253	0.05	0.05	0.86	0.85	0.80	0.40	0.00	0.00	0.76	0.27	0.25	0.01	0.81	0.23	0.56	0.41
254	0.02	0.06	1.00	0.27	0.78	0.21	0.00	0.00	NaN	0.00	0.00	0.00	0.03	1.00	0.04	0.03
257	0.01	0.03	0.70	0.64	1.00	0.64	0.00	0.00	1.00	0.09	0.00	0.00	0.05	0.45	0.25	0.21
258	0.01	0.01	0.70	0.70	0.88	0.39	0.00	0.00	0.79	0.32	0.25	0.01	0.82	0.25	0.49	0.35
259	0.01	0.01	0.68	0.68	0.61	0.34	0.00	0.00	0.59	0.21	0.25	0.01	0.82	0.24	0.58	0.47
260	0.00	0.00	0.52	0.48	0.75	0.31	0.00	0.00	0.75	0.10	0.00	0.00	0.05	0.86	0.26	0.17
261	0.00	0.00	0.50	0.48	0.63	0.30	0.00	0.00	0.33	0.06	0.00	0.00	0.01	0.15	0.14	0.09
262	0.01	0.03	0.89	0.24	0.78	0.21	0.00	0.00	NaN	0.00	0.00	0.00	0.03	1.00	0.20	0.06
265	0.00	0.00	0.48	0.45	0.75	0.31	0.00	0.00	0.75	0.10	0.00	0.00	0.05	0.86	0.22	0.14
266	0.00	0.00	0.50	0.48	0.67	0.36	0.00	0.00	0.33	0.06	0.00	0.00	0.01	0.15	0.14	0.09
301	0.48	0.79	0.96	0.80	0.83	0.31	0.10	0.07	0.74	0.64	1.00	0.13	0.94	0.25	0.42	0.38
302	0.31	0.65	0.97	0.67	0.97	0.65	0.14	0.27	0.62	0.48	1.00	0.17	1.00	0.58	0.37	0.33
303	0.40	0.82	0.80	0.82	0.89	0.80	0.04	0.29	0.51	0.53	1.00	0.18	0.93	0.80	0.41	0.49
304	0.71	0.95	0.97	0.96	0.95	0.96	0.11	0.26	0.75	0.70	0.85	0.22	0.91	0.91	0.74	0.66
H-means	0.45	0.61	0.91	0.89	0.90	0.69	0.08	0.24	0.92	0.72	0.81	0.18	0.35	0.70	0.80	0.74

Table 8.5: Full results

Chapter 9

OAEI-2005: lesson learned and improvements

Beside the results of the evaluation properly speaking, there is a number of lessons that can be taken from running it. We consider below a number of them before providing some future plans linked to these remarks.

9.1 Lesson learned

From the 2005 OAEI campaign we can draw the following lessons:

More tools It seems that there are more and more tools able to jump in this kind of tests. This is a measure of the increase in interoperability of the tools developed for matching ontologies. This is also a call for carrying on these experiments (they are possible and people participate).

Tool robustness Contrary to last year it seems that the tools are more robusts and people deal with more wider implementation of OWL. However, this can be that we tuned the tests so that no one has problems. But our global impression is that both tools and the way people design OWL ontologies have improved.

Few suited corpus Contrary to what many people think, it is not that easy to find ontological corpora suitable for this evaluation test. From the proposals we had from last year, only one proved to be usable and with great difficulty (on size, conformance and juridical aspects). One could claim that matching thus solve no problem at all or that we do not yet have developed ontologies of significant size that people are ready to release.

Test realism The extension of the benchmark tests towards more coverage of the space is relatively systematic. However, it would be interesting and certainly more realistic, instead of crippling all names to do it for some random proportion of them (5% 10% 20% 40% 60% 100% random change). This has not been done for reason of time.

Size problems The real world benchmarks were huge benchmarks. Two different strategies have been taken with them: cutting them in a huge set of tiny benchmark or providing them as is. The first solution brings us away from "real world", while the second one raised serious

problems to the participants. It would certainly be worth designing these tests in order to assess the current limitation of the tools by providing an increasingly large sequence of such tests (0.1%, 1%, 10%, 100% of the corpus for instance).

Difficult validation Validation of the results is quite difficult to establish. Problems for evaluating the directory test have been mentioned as well as problems in evaluating the results of semantic matchers whose goal is correctness and completeness rather than precision and recall. These measures are related but not equivalent. For dealing with these problems which are typically semantic problems, measures that take semantic into account must be developed.

9.2 Future plans

In order to address these problems, several number actions can be taken and will be considered for future evaluations:

Real real world example This first measure has been suggested by one of the participant at the workshop. Indeed, the real world tests used this year can be criticised for not being totally natural: one of them split huge ontologies in pieces and the other one changed the ontology language. Moreover, their evaluation is difficult. One way to reduce this problem would be to ask someone with real problems, with a real interest to see ontology matching at work to submit the problem and to evaluate it (or to provide the criterion). This would have the advantage of some test case not made by researchers (so less suspect to bias) and solving a real problem. For that purpose, we proposed to find some interested party, preferably from the industry sector, with an ontology matching need, to provide ontologies and to evaluate the results in function of its problem. A call has been posted on the OAEI website.

New measures and evaluation techniques Since last year we made some progress in evaluation techniques (in particular with the computation of precision/recall graphs). However, the results are still not satisfying. Thus we are working on providing better evaluation measures and methodologies. A number of these have already been investigated in depth and are presented in the next part of this document.

Sampling tests It becomes clear that if we want to assess the scalability of the proposed methods, it would be very useful to propose versions of the tests of different size. In particular, this will be done with particularly large ontologies. It may also be useful to have some randomness in the systematically generated tests of the benchmark suite. So we will work toward this goal.

Part III

Evaluation Issues

Chapter 10

Measures

10.1 Introduction

In order to evaluate the performance of matching algorithms it is necessary to confront them with ontologies to match and to compare the results based on some criterion. The most prominent criteria are precision and recall originating from information retrieval and adapted to the matching task. Precision and recall are based on the comparison of the resulting alignment A with another standard alignment R, effectively comparing which correspondences are found and which are not. These criteria are well understood and widely accepted.

However, as we have experienced in last year's Ontology Alignment Contest [Sure *et al.*, 2004], they have the drawback to be of the all-or-nothing kind. An alignment may be very close to the expected result and another quite remote from it and both return the same precision and recall. The reason for this is that the criteria only compare two sets of correspondences without considering if these are close or remote to each other: if they are not the same exact correspondences, they score zero. They both score identically low, despite their different quality. It may be helpful for users to know whether the found alignments are close to the expected one and easily repairable or not. It is thus necessary to measure the proximity between alignments instead of their strict equality.

In this chapter we investigate some measures that generalize precision and recall in order to overcome the problems presented above. We reproduce here the main part of [Ehrig and Euzenat, 2005]. We first provide the basic definitions of alignments, precision and recall as well as a motivating example (§10.2). We then present a framework for generalizing precision and recall (§10.3). This framework is instantiated by four different measures (including classical precision and recall) (§10.4) and we show on the motivating example that the proposed measures do not exhibit the rigidity of classical precision and recall (§10.5).

10.2 Foundations

10.2.1 Alignment

We consider the result of matching, called alignment, as a set of pairs of entities $\langle e, e' \rangle$ from two ontologies O and O' that are supposed to satisfy a certain relation r with a certain confidence n.

Definition 5 (Alignment, correspondence)**.** *Given two ontologies O and O', an alignment between O and O' is a set of correspondences (i.e., 4-uples): $\langle e, e', r, n\rangle$ with $e \in O$ and $e' \in O'$ being the two matched entities, r being a relationship holding between e and e', and n expressing the level of confidence $[0..1]$ in this correspondence.*

A matching algorithm returns an alignment A which is compared with a reference alignment R. Let us illustrate this through a simple example. Figure 10.1 presents two ontologies together with two alignments A_1 and R. In this example, for the sake of simplification, the relation is always '=' and the confidence is always 1.0.

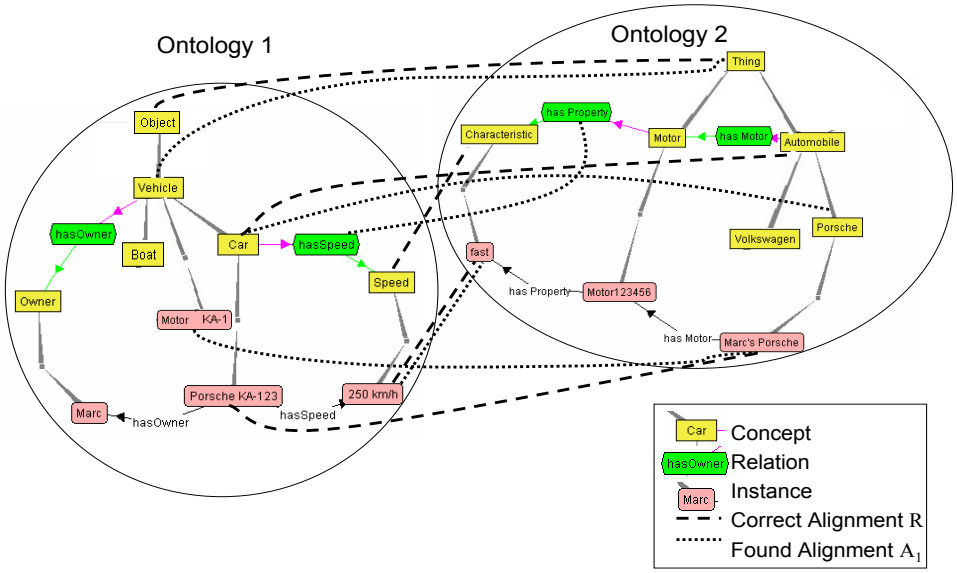

Figure 10.1: Two Aligned Ontologies

The alignment A_1 is defined as follows:

```
<o1:Vehicle,o2:Thing,=,1.0>
<o1:Car,o2:Porsche,=,1.0>
<o1:hasSpeed,o2:hasProperty,=,1.0>
<o1:MotorKA1,o2:MarcsPorsche,=,1.0>
<o1:250kmh,o2:fast,=,1.0>
```

We present another reasonable alignment A_2:

```
<o1:Car,o2:Thing,=,1.0>
<o1:hasSpeed,o2:hasProperty,=,1.0>
<o1:MotorKA1,o2:MarcsPorsche,=,1.0>
<o1:250kmh,o2:fast,=,1.0>
```

and an obviously wrong alignment A_3:

```
<o1:Object,o2:Thing,=,1.0>
<o1:Owner,o2:Volkswagen,=,1.0>
<o1:Boat,o2:Porsche,=,1.0>
<o1:hasOwner,o2:hasMotor,=,1.0>
<o1:Marc,o2:fast,=,1.0>
```

Further, we have the following reference alignment (R):

```
<o1:Object,o2:Thing,=,1.0>
<o1:Car,o2:Automobile,=,1.0>
<o1:Speed,o2:Characteristic,=,1.0>
<o1:250kmh,o2:fast,=,1.0>
<o1:PorscheKA123,o2:MarcsPorsche,=,1.0>
```

10.2.2 Precision and Recall

The usual approach for evaluating the returned alignments is to consider them as sets of correspondences and check for the overlap of the two sets. This is naturally obtained by applying the classical measure of precision and recall [van Rijsbergen, 1979], which are the ratio of the number of true positive ($|R \cap A|$) on that of the retrieved correspondences ($|A|$) and those expected ($|R|$) respectively.

Definition 6 (Precision, Recall)**.** *Given a reference alignment R, the precision of some alignment A is given by*

$$P(A, R) = \frac{|R \cap A|}{|A|}$$

and recall is given by

$$R(A, R) = \frac{|R \cap A|}{|R|}.$$

10.2.3 Problems with Current Measures

These criteria are well understood and widely accepted. However, they have the drawback that whatever correspondence has not been found is definitely not considered. As a result, they do not discriminate between a bad and a better alignment and they do not measure the user effort required to correct alignments.Indeed, it often makes sense to not only have a decision whether a particular correspondence has been found or not, but somehow measure the proximity of the found alignments. This implies that "near misses" are also taken into consideration instead of only the exact matches. As a matter of example, it will be clear to anybody that among the alignments presented above, A_3 is not a very good alignment and A_1 and A_2 are better alignments. However, they score almost exactly the same in terms of precision (.2) and recall (.2). Moreover, the alignments will have to go through user scrutiny and correction before being used. It is worth measuring the effort required by the user for correcting the provided alignment instead of only if some correction is needing. This also calls for a relaxation of precision and recall.

10.3 Generalizing Precision and Recall

As precision and recall are easily explained measures, it is good to extend them. This also ensures that measures derived from precision and recall (e.g., F-measure) still can be computed easily. For these reasons, we propose to generalize these measures. In fact, if we want to generalize precision and recall, we should be able to measure the proximity of alignment sets rather than the strict size of their overlap. Instead of taking the cardinal of the intersection of the two sets ($|R \cap A|$), the natural generalizations of precision and recall measure their proximity ($\omega(A, R)$).

Definition 7 (Generalized precision and recall)**.** *Given a reference alignment R and an overlap function ω between alignments, the precision of an alignment A is given by*

$$P_\omega(A, R) = \frac{\omega(A, R)}{|A|}$$

and recall is given by

$$R_\omega(A, R) = \frac{\omega(A, R)}{|R|}.$$

10.3.1 Basic properties

In order, for these new measures to be true generalizations, we would like ω to share some properties with $|R \cap A|$. In particular, the measure should be positive:

$$\forall A, B, \omega(A, B) \geq 0 \qquad \text{(positiveness)}$$

and should not exceed the minimal size of both sets:

$$\forall A, B, \omega(A, B) \leq min(|A|, |B|) \qquad \text{(maximality)}$$

Further, this measure should only add more flexibility to the usual precision and recall so their values cannot be worse than the initial evaluation:

$$\forall A, B, \omega(A, B) \geq |A \cap B| \qquad \text{(boundedness)}$$

Hence, the main constraint faced by the proximity is the following:

$$|A \cap R| \leq \omega(A, R) \leq min(|A|, |R|)$$

This is indeed a true generalization because, $|A \cap R|$ satisfies all these properties. One more property satisfied by precision and recall that we will not enforce here is symmetry. This guarantees that the precision and recall measures are true normalized similarities.

$$\forall A, B, \omega(A, B) = \omega(B, A) \qquad \text{(symmetry)}$$

We will not require symmetry, especially since A and R are not in symmetrical positions.

10.3.2 Designing Overlap Proximity

There are many different ways to design a proximity between two sets satisfying these properties. The most obvious one, that we retain here, consists of finding correspondences matching each other and computing the sum of their proximity. This can be defined as an overlap proximity:

Definition 8 (Overlap proximity)**.** *A measure that would generalize precision and recall is:*

$$\omega(A,R) = \sum_{\langle a,r\rangle \in M(A,R)} \sigma(a,r)$$

in which $M(A,R)$ is a matching between the correspondences of A and R and $\sigma(a,r)$ a proximity function between two correspondences.

The standard measure $|A \cap R|$ used in precision and recall is such an overlap proximity which provides the value 1 if the two correspondences are equal and 0 otherwise. There are two tasks to fulfill when designing such an overlap proximity function:

- the first one consists of designing the correspondence matching M;
- the second one is to define a proximity measure σ on correspondences.

We consider these two issues below.

10.3.3 Matching Correspondences

A matching between alignments is a set of correspondence pairs, i.e., $M(A,R) \subseteq A \times R$. However, if we want to keep the analogy with precision and recall, it will be necessary to restrict ourselves to the matchings in which an entity from the ontology does not appear twice, i.e., $|M(A,R)| \leq min(|A|,|R|)$. This is compatible with precision and recall for two reasons: (i) in these measures, any correspondence is identified only with itself, and (ii) appearing more than once in the matching would not guarantee that the resulting measure is bounded by 1 . The natural choice is to select the best match because this guarantees that this function generalizes precision and recall. There are $\frac{|A|!}{(|A|-|R|)!}$ candidate matches (if $|A| \geq |R|$). The natural choice is to select the best match because this guarantees that the function generalizes precision and recall.

Definition 9 (Best match)**.** *The best match $M(A,R)$ between two sets of correspondences A and R, is the subset of $A \times R$ which maximizes the overall proximity and in which each element of A (resp. R) belongs to only one pair:*

$$M(A,R) \in Max_{\omega(A,R)}\{M \subseteq A \times R\}$$

As defined here, this best match is not unique. This is not a problem for our purpose because we only want to find the highest value for ω and any of these best matches will yield the same value. Of course, the definition M and ω are dependent of each other, but this does not prevent from computing them. They are usually computed together but presenting them separately is clearer.

10.3.4 Correspondence Proximity

In order to compute $\omega(A, R)$, we need to measure the proximity between two matched correspondences (i.e., $\langle a, r\rangle \in M(A, R)$) on the basis of how close the result is to the ideal one. Each element in the tuple $a = \langle e_a, e'_a, r_a, n_a\rangle$ will be compared with its counterpart in $r = \langle e_r, e'_r, r_r, n_r\rangle$. For any two correspondences (the found a and the reference r) we compute three similarities σ_{pair}, σ_{rel}, and σ_{conf}. If elements are identical, correspondence proximity has to be 1 (maximality). If they differ, proximity is lower, always according to the chosen strategy. In contrast to the standard definition of similarity, the mentioned proximity measures do not necessarily have to be symmetric. We will only consider normalized proximities, i.e., measures whose value ranges within the unit interval $[0\ 1]$, because this is a convenient way to guarantee that

$$\sigma(A, R) \leq min(|A|, |R|)$$

The component proximity measure is defined in the following way:

$\sigma_{pair}(\langle e_a, e_r\rangle, \langle e'_a, e'_r\rangle)$**:** How is one entity pair similar to another entity pair? In ontologies we can in principal follow any relation which exists (e.g., subsumption, instantiation), or which can be derived in a meaningful way. The most important parameters are the relations to follow and their effect on the proximity.

$\sigma_{rel}(r_a, r_r)$**:** Often the alignment relations are more complex, e.g., represent subsumption, instantiation, or compositions. Again, one has to assess the similarity between these relations. The two relations of the alignment cell can be compared based on their distance in a conceptual neighborhood structure [Euzenat *et al.*, 2003, Freksa, 1992].

$\sigma_{conf}(n_a, n_r)$**:** Finally, one has to decide, what to do with different levels of confidence. The similarity could simply be the difference. Unfortunately, none of the current alignment approaches have an explicit meaning attached to confidence values, which makes it rather difficult in defining an adequate proximity.

Once these proximities are established, they have to be aggregated. The constraints on the aggregation function ($Aggr$) are:

- normalization preservation (if $\forall i, 0 \leq c_i \leq 1$ then $0 \leq Aggr_i c_i \leq 1$);
- maximality (if $\forall i, c_i = 1$ then $Aggr_i c_i = 1$);
- local monotonicity (if $\forall i \neq j, c_i = c'_i = c''_j$ and $c_j \leq c'_j \leq c''_j$ then $Aggr_i c_i \leq Aggr_i c'_i \leq Aggr_i c''_i$).

Here, we consider aggregating them through multiplication without further justification. Other aggregations (e.g., weighted sum) are also possible.

Definition 10 (Correspondence proximity)**.** *Given two correspondences* $\langle e_a, e'_a, r_a, n_a\rangle$ *and* $\langle e_r, e'_r, r_r, n_r\rangle$*, their proximity is:*

$$\sigma(\langle e_a, e'_a, r_a, n_a\rangle, \langle e_r, e'_r, r_r, n_r\rangle) =$$

$$\sigma_{pair}(\langle e_a, e_r\rangle, \langle e'_a, e'_r\rangle) \times \sigma_{rel}(r_a, r_r) \times \sigma_{conf}(n_a, n_r)$$

We have provided constraints and definitions for M, ω, and σ. We now turn to concrete measures.

10.4 Concrete Measures

From this simple set of constraints, we have designed several concrete measures:

symmetric is a simple measure of the distance in the ontologies between the found entities and the reference one;

edit measures the effort necessary to modify the errors found in the alignments;

oriented is a specific measure which uses different ω for precision and recall depending on the impact an error has on these measures, e.g., when one wants to retrieve instances of some class, a subclass of the expected one is correct but not complete, it thus affects recall but not precision.

We consider four cases of relaxed precision and recall measures based on the above definitions. We first give the definition of usual precision and recall within this framework.

10.4.1 Standard Precision and Recall

For standard precision and recall, the value of ω is $|A \cap R|$. This is indeed an instance of this framework, if the proximity used is based on the strict equality of the components of correspondences.

Definition 11 (Equality proximity)**.** *The equality proximity is characterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) = \begin{cases} 1 & \text{if } \langle e_a, e'_a\rangle = \langle e_r, e'_r\rangle \\ 0 & \text{otherwise} \end{cases}$$

$$\sigma_{rel}(r_a, r_r) = \begin{cases} 1 & \text{if } r_a = r_r \\ 0 & \text{otherwise} \end{cases}$$

$$\sigma_{conf}(n_a, n_r) = \begin{cases} 1 & \text{if } n_a = n_r \\ 0 & \text{otherwise} \end{cases}$$

In the measure used for the EON-2004 contest of last year the theoretical[1] measure to be be used was:

Definition 12 (EON proximity)**.** *The proximity used for EON-2004 is characterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) = \begin{cases} 1 & \text{if } \langle e_a, e'_a\rangle = \langle e_r, e'_r\rangle \\ 0 & \text{otherwise} \end{cases}$$

$$\sigma_{rel}(r_a, r_r) = \begin{cases} 1 & \text{if} \quad r_a = r_r \\ .5 & \text{if} \quad r_a = \leq \text{ and } r_r == \text{ or } r_a = \geq \text{ and } r_r == \text{ or} \\ & \quad\; r_a == \text{ and } r_r = \leq \text{ or } r_a == \text{ and } r_r = \geq \\ 0 & \quad\; \text{otherwise} \end{cases}$$

$$\sigma_{conf}(n_a, n_r) = \begin{cases} 1 & \text{if } n_a \neq 0 \text{ and } n_r \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

It already introduced some tolerance for algorithms unable to compute subsumption relationships and retained all the correspondence with a non zero confidence as fully confident. As a result, the values were already a bit weakened.

[1] In fact, this is theoretical because the relaxed relation equality has not been computed

10.4.2 Symmetric Proximity

The easiest way to relax precision and recall is to have some distance δ on the elements in ontologies and to weight the proximity with the help of this distance: the higher the distance between two entities in the matched correspondences, the lower their proximity. This can be defined as:

$$\left.\begin{array}{rl} & \delta(e_a, e_r) \leq \delta(e_b, e_r) \\ \text{and} & \delta(e'_a, e'_r) \leq \delta(e'_b, e'_r) \end{array}\right\}$$

$$\Longrightarrow \sigma(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) \geq \sigma(\langle e_b, e'_b\rangle, \langle e_r, e'_r\rangle)$$

As a simple example of such a symmetric similarity, we use a distance in which a class is at distance 0 of itself, at distance 0.5 of its direct sub- and superclasses, and at a distance 1 of any other class. This could be further refined by having a similarity inversely proportional to the distance in the subsumption tree. Likewise, this similarity may also be applied to properties and instances (through part-of relationships in the latter case). The similarity between pairs is the complement of these similarities The result is displayed in Table 10.1. We always mention the assumed alignment and the actual correct alignment.

found e,e'	closest correct e,e'	similarity σ_{pair}	comment
e,e'	e,e'	1	correct correspondence
c,c'	c,$sup(c')$	0.5	returns more specialized instances
c,c'	$sup(c)$,c'	0.5	returns more general instances
c,c'	c,$sub(c')$	0.5	returns more general instances
c,c'	$sub(c)$,c'	0.5	returns more specialized instances
r,r'	r,$sup(r')$	0.5	returns more spec. relation instances
r,r'	$sup(r)$,r'	0.5	returns more gen. relation instances
r,r'	r,$sub(r')$	0.5	returns more gen. relation instances
r,r'	$sub(r)$,r'	0.5	returns more spec. relation instances
i,i'	i,$super(i')$	0.5	returns a more restricted instance
i,i'	$super(i)$,i'	0.5	returns a too broad instance
i,i'	i,$sub(i')$	0.5	returns a too broad instance
i,i'	$sub(i)$,i'	0.5	returns a more restricted instance

Table 10.1: Similarities based on Entity Pairs

Table 10.2 consider the proximity between relations. It only presents the similarity between equality (=) and other relations.

For the confidence distance we simply take the complement of the difference. The final precision is calculated according to the formula presented in the previous section:

Definition 13 (Symmetric proximity)**.** *The symmetric proximity is characterized by:*

$$\begin{aligned} \sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) & \textit{ as defined in Table 10.1} \\ \sigma_{rel}(r_a, r_r) & \textit{ as defined in Table 10.2} \\ \sigma_{conf}(n_a, n_r) = 1 - |n_a - n_r|. & \end{aligned}$$

found relation	correct relation	similarity σ_{rel}	comment
$e = e'$	$e = e'$	1	correct relation
$c = c'$	$c \subset c'$	0.5	returns more instances than correct
$c = c'$	$c \supset c'$	0.5	returns less instances than possible, but these are correct
$r = r'$	$r \subset r'$	0.5	
$r = r'$	$r \supset r'$	0.5	
$i = i'$	i partOf i'	0.5	
$i = i'$	i consistsOf i'	0.5	

Table 10.2: Similarities based on Relations

10.4.3 Measuring Correction Effort

If users have to check and correct alignments, the quality of alignment algorithms can be measured through the effort required for transforming the obtained alignment into the (correct) reference one [Do *et al.*, 2002].

This measure can be implemented as an edit distance [Levenshtein, 1966]: an edit distance defines a number of operations by which an object can be corrected (here the the operations on correspondences authorized) and assigns a cost to each of these operations (here the effort required to identify and repair some mistake). The cost of a sequence of operations is the sum of their cost and the distance between two objects is the cost of the less costly sequence of operations that transform one object into the other one. The result can always be normalized in function of the size of the largest object. Such a distance can be turned into a proximity by taking its complement with regard to 1.

Table 10.3 provides such plausible weights. Usually classes are organized in a taxonomy in which they have less direct super- than subclasses. It is thus easier to correct a class to (one of) its superclass than to one of its subclasses. As a consequence, the proximity is dissymmetric. Such a measure should also add some effort when classes are not directly related, but this has not been considered here.

The edit distance between relations is relatively easy to design since, generally, changing from one relation to another can be done with just one click. Thus, the relational similarity equals 1 if the relations are the same and 0.5 otherwise. In this correction effort measure, the confidence factor does not play an important role: ordering the correspondences can only help the user to know that after some point she will have to discard many correspondences. We thus decided to not take confidence into account and thus, their proximity will always be 1.

Definition 14 (Effort-based proximity)**.** *The effort-based proximity is characterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) \textit{ as defined in Table 10.3}$$

$$\sigma_{rel}(r_a, r_r) = \begin{cases} 1 & \textit{if } r_a = r_r \\ 0.5 & \textit{otherwise} \end{cases}$$

$$\sigma_{conf}(n_a, n_r) = \begin{cases} 1 & \textit{if } n_a \neq 0 \textit{ and } n_r \neq 0 \\ 0 & \textit{otherwise} \end{cases}$$

found e,e'	closest correct e,e'	effort	similarity σ_{pair}	comment
e,e'	e,e'	0	1	correct alignment
c,c'	c,$sup(c')$	0.4	0.6	returns more spec. instances
c,c'	$sup(c)$,c'	0.4	0.6	returns more gen. instances
c,c'	c,$sub(c')$	0.6	0.4	returns more gen. instances
c,c'	$sub(c)$,c'	0.6	0.4	returns more spec. instances
r,r'	r,$sup(r')$	0.4	0.6	
r,r'	$sup(r)$,r'	0.4	0.6	
r,r'	r,$sub(r')$	0.6	0.4	
r,r'	$sub(r)$,r'	0.6	0.4	
i,i'	i,$super(i')$	0.4	0.6	returns a more restricted inst.
i,i'	$super(i)$,i'	0.4	0.6	returns a too broad inst.
i,i'	i,$sub(i')$	0.6	0.4	returns a too broad inst.
i,i'	$sub(i)$,i'	0.6	0.4	returns a more restricted inst.

Table 10.3: Effort-based proximity between Entity Pairs

To be accurate, such an effort proximity would have been better aggregated with an additive and normalized aggregation function rather than multiplication.

10.4.4 Precision- and Recall-oriented Measures

One can also decide to use two different similarities depending on their application for evaluating either precision or recall. We here provide two such measures and justify the given weights. Precision is normally a measure of accuracy i.e., the returned results need to be correct. Every wrong result will therefore entail a penalty. We assume the user poses a query to the system as follows: "return me all instances of e". The system then returns any instance corresponding to the alignment i.e. e'. Vice versa, for the relaxed recall we want to avoid missing any correct result. This affects the similarity relations and weights.

Relaxed Precision

In Table 10.4 and 10.5 we present the precision similarity for pairs and relations. The comments in each line explain the decision for the weights.

For the distance within the confidence we again use the complement of the difference.

Definition 15 (Precision-oriented proximity)**.** *The precision-oriented proximity is characterized by:*

$$\begin{aligned}\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) &\text{ as defined in Table 10.4}\\ \sigma_{rel}(r_a, r_r) &\text{ as defined in Table 10.5}\\ \sigma_{conf}(n_a, n_r) &= 1 - |n_a - n_r|.\end{aligned}$$

Relaxed Recall

In Table 10.6 and 10.7 we present the recall similarity for pairs and relations. Basically many distances are just mirrored compared to the precision case.

The final recall is computed as usual:

found e,e'	closest correct e,e'	similarity σ_{pair}	comment
e,e'	e,e'	1	correct correspondence
c,c'	c,$sup(c')$	1	returns more specialized instances, these are correct
c,c'	$sup(c)$,c'	0.5	returns more general instances, includes some correct results
c,c'	c,$sub(c')$	0.5	returns more general instances, includes some correct results
c,c'	$sub(c)$,c'	1	returns more specialized instances, these are correct
r,r'	r,$sup(r')$	1	
r,r'	$sup(r)$,r'	0.5	
r,r'	r,$sub(r')$	0.5	
r,r'	$sub(r)$,r'	1	
i,i'	i,$super(i')$	0.5	returns a more restricted instance
i,i'	$super(i)$,i'	0	returns a too broad instance
i,i'	i,$sub(i')$	0	returns a too broad instance
i,i'	$sub(i)$,i'	0.5	returns a more restricted instance

Table 10.4: Similarities for Relaxed Precision based on Entity Pairs

found relation	correct relation	similarity σ_{rel}	comment
$e = e'$	$e = e'$	1	correct relation
$c = c'$	$c \subset c'$	0.5	returns more instances than correct
$c = c'$	$c \supset c'$	1	returns less instances than possible, but these are correct
$r = r'$	$r \subset r'$	0.5	
$r = r'$	$r \supset r'$	1	
$i = i'$	i partOf i'	0.5	
$i = i'$	i consistsOf i'	1	

Table 10.5: Similarities for Relaxed Precision based on Relations

Definition 16 (Recall-oriented proximity)**.** *The recall-oriented proximity is characterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) \text{ as defined in Table 10.6}$$
$$\sigma_{rel}(r_a, r_r) \text{ as defined in Table 10.7}$$
$$\sigma_{conf}(n_a, n_r) = 1 - |n_a - n_r|.$$

10.5 Example

In the introduction of the chapter we have presented a pair of ontologies, the reference alignment, and three different identified alignments. We will now apply the different proposed precision and recall measures to these example alignments. Please note that they mainly illustrate entity pair similarities, as relations and confidences are always identical. Table 10.8 provides the results. For the oriented measure we assume that the query is given in ontology 1 and the answer has to be retrieved in ontology 2. As the oriented measure is dissymmetric, one has to define this direction

found e,e'	closest correct e,e'	similarity σ_{pair}	comment
e,e'	e,e'	1	correct correspondence
c,c'	c,$sup(c')$	0.5	returns more specialized instances, misses some
c,c'	$sup(c)$,c'	1	returns more general instances, includes the correct results
c,c'	c,$sub(c')$	1	returns more general instances, includes the correct results
c,c'	$sub(c)$,c'	0.5	returns more specialized instances, misses some
r,r'	r,$sup(r')$	0.5	
r,r'	$sup(r)$,r'	1	
r,r'	r,$sub(r')$	1	
r,r'	$sub(r)$,r'	0.5	
i,i'	i,$super(i')$	0	returns a more restricted instance, misses correct
i,i'	$super(i)$,i'	0.5	returns a broader instance
i,i'	i,$sub(i')$	0.5	returns a broader instance
i,i'	$sub(i)$,i'	0	returns a more restricted instance, misses correct

Table 10.6: Similarities for Relaxed Recall based on Entity Pairs

found relation	correct relation	similarity σ_{rel}	comment
$e = e'$	$e = e'$	0	correct relation
$c = c'$	$c \subset c'$	0	returns more instances than correct
$c = c'$	$c \supset c'$	0.5	returns less instances than possible, misses some
$r = r'$	$r \subset r'$	0	
$r = r'$	$r \supset r'$	0.5	
$i = i'$	i partOf i'	0	
$i = i'$	i consistsOf i'	0.5	

Table 10.7: Similarities for Relaxed Recall based on Relations

beforehand.

ω	(R,R)		(R,A_1)		(R,A_2)		(R,A_3)	
	P	R	P	R	P	R	P	R
standard	1.0	1.0	0.2	0.2	0.25	0.2	0.2	0.2
symmetric	1.0	1.0	0.4	0.4	0.375	0.3	0.2	0.2
edit	1.0	1.0	0.44	0.44	0.35	0.28	0.2	0.2
oriented	1.0	1.0	0.5	0.5	0.375	0.4	0.2	0.2

Table 10.8: Precision recall result on the alignments of Figure 10.1

The measures which have been introduced address the problems raised in the introduction and fulfill the requirements:

- They keep precision and recall untouched for the best alignment (R);

- They help discriminating between irrelevant alignments (A_3) and not far from target ones (A_1 and A_2);
- Specialized measures are able to emphasize some characteristics of alignments: ease of modification, correctness or completeness. For instance, let's consider the oriented measures. In our example A_1 has two very near misses, which leads to a relatively high precision. In A_2 however the miss is bigger, but by aligning one concept to its superconcept recall rises relatively to precision.

These results are based on only one example. They have to be systematized in order to be extensively validated. Our goal is to implement these measures within the Alignment API and to use them on the forthcoming results of the Ontology Alignment Evaluation 2005[2] in order to have real data on which the relevance of the proposed measures can be more openly debated.

10.6 Related Work

The naturally relevant work is [Do *et al.*, 2002] which has considered precisely the evaluation of schema matching. However, the authors only note the other mentioned problem (having two measures instead of one) and use classical aggregation (overall and F-measure) of precision and recall. In computational linguistics, and more precisely multilingual text alignment, [Langlais *et al.*, 1998] has considered extending precision and recall. Their goal is the same as ours: increasing the discriminating power of the measures. In this work, the mathematical formulation is not changed but the granularity of compared sets changes: instead of comparing sentences in a text, they compare words in sentences in a text. This helps having some contribution to the measures when most of the words are correctly aligned while the sentences are not strictly aligned.

In the Alignment API [Euzenat, 2004], there is another evaluation measure which directly computes a distance based on a weighted symmetric difference (weights are the confidences of each correspondence in the alignment). This measure could be used in the generalization proposed here (the distance would then be based on confidence difference and would generally satisfy $P'(A,R) \leq P(A,R)$ and $R'(A,R) \leq R(A,R)$. The deeper proposal for extending precision and recall comes from hierarchical text categorization in which texts are attached to some category in a taxonomy [Sun and Lin, 2001]. Usually, texts are attached to the leaves, but when algorithms attach them to the intermediate categories, it is useful to discriminate between a category which is irrelevant and a category which is an immediate super category of the expected one. For that purpose, they introduce an extension of precision (recall is redefined similarly) such that:

$$P_{CS} = \frac{max(0, |A \cap R| + FpCon + FnCon)}{|A| + FnCon}$$

in which $FpCon$ (resp. $FnCon$) is the contribution to false positive (resp. false negative), i.e., the way incorrectly classified documents could contribute to its incorrect category anyway. The maximization is necessary to prevent the result from being negative (because the contribution is defined with respect to the average such contribution). The contribution is measured in two ways. The first one is a category similarity that is computed on the features of categories (categories and documents are represented by a vector of features and the membership to some category is based

[2] `http://oaei.inrialpes.fr/2005/`

on a distance between these vectors). The second one is based on the distance between categories in the taxonomy.

This measure does not seem to be a generalization of standard precision and recall as the one presented here. In particular, because the contributions can be negative, this measure can be lower than standard precision and recall. The idea of retracting the contribution from wrongly classified documents is not far from the idea developed here. However, the computation of this contribution with regard to some average and the addition of some contribution to the divisor do not seem justified.

10.7 Discussion

Evaluation of matching results is often made on the basis of the well-known and well-understood precision and recall measures. However, these measures do not discriminate accurately between methods which do not provide the exact results. In the context where the result of alignments have to be screened by humans, this is an important need. In order to overcome the lack of discrimination affecting precision and recall, we provided a framework properly generalizing these measures (in particular, precision and recall can be expressed in this framework). We have presented the general principles that guide the design of such generalizations.

The framework has been instantiated in three different measures, each one aiming at favoring some particular aspects of alignment utility. We show that these measures indeed avoid the shortcomings of standard evaluation criteria. The proposed measures were having the expected results:

- they keep precision and recall untouched for the best alignment;
- they help discriminating between irrelevant alignments and not far from target ones;
- specialized measures are able to emphasize some characteristics of alignments: ease of modification, correctness or completeness.

They should however, be further investigated in order to find better formulations: more discrepancy needs to be considered, more progressive distance (e.g., not direct subclasses) and rationalized design of weights. The measures have been implemented in the Alignment API [Euzenat, 2004], which has been used for evaluation at the OAEI.

This generalization framework is not the only possible one since we have made a number of choices:

- on the form of the alignment similarity (Definition 8);
- on the kind of alignment matching (Definition 9);
- on the form of the correspondence similarity (Definition 10).

More work has to be done in order to assess the potential of other choices in these functions. The most important work is to consider these proposed measures in real evaluation of alignment systems and to identify good measures for further evaluations. These measures have been implemented within the Alignment API [Euzenat, 2004] and processed the results of the Ontology

Alignment Evaluation 2005. Unfortunatelly, this does not change the results we are currently investigating if this is due to an artefact of the test set or of our implementation of the measures.

Another development currently under investigation consists of developing similar measures accounting for the semantics of the language used for ontologies. This would solve the problems that have been noted during the 2005 evaluation.

Chapter 11

Generation of Reference Mappings

One of the problem we are faced when designing test cases for evaluation is that of acquiring the reference alignments. Up to now the acquisition of the reference mappings that hold among the nodes of two taxonomies is performed manually. Similarly to the annotated corpora for information retrieval or information extraction, a corpus of pairwise relationships is annotated. Of course such an approach prevents the opportunity of having large corpora. The number of mappings between two taxonomies are quadratic with respect to taxonomy size, what makes hardly possible the manual mapping of real world size taxonomies. It is worthwhile to remember that web directories, for example, have tens thousands of nodes. Certain heuristics can help in reducing the search space but the human effort is still too demanding. This is our goal here to provide such a method in order to design decent test sets.

Our proposal is to build a reference interpretation for a node looking at its use. We argue that the semantics of nodes can be derived by their pragmatics, i.e., how they are used. In context of web directories, the nodes of a taxonomy are used to classify documents. The set of documents classified under a given node implicitly defines its meaning. This approach has been followed by other researchers. For example in [Doan *et al.*, 2003b, Ichise *et al.*, 2003] the interpretation of a node is approximated by a model computed through statistical learning. Of course the accuracy of the interpretation is affected by the error of the learning model. We follow a similar approach but without the statistical approximation. Our working hypothesis is that the meaning of two nodes is equivalent if the sets of documents classified under those nodes have a meaningful overlap. The basic idea is to compute the relationship hypotheses based on the co-occurence of documents. This document-driven interpretation can be used as a reference value for the evaluation of competing matching solutions. A simple definition of equivalence relationship based on documents can be derived by the F1 measure of information retrieval.

Figure 11.1 shows a simple example. In the graphical representation we have two taxonomies, for each of them we focus our attention on a reference node. Let be S and P two sets of documents classified under the reference nodes of the first and second taxonomies respectively. We will refer to A_S and A_P as the set of documents classified under the ancestor nodes of S and P. Conversely, we will refer to T_S and T_P as the set of documents classified under the subtrees of S and P. The goal is to define a relationship hypothesis based on the overlapping of the set of documents, i.e. the pragmatic use of the nodes.

The first step, the *equivalence* relationship, can be easily formulated as the F1 measure of information retrieval [Baeza-Yates and Ribeiro-Neto, 1999]. The similarity of two sets of documents is defined as the ratio between the marginal sets and the shared documents:

$$Equivalence = \frac{|O_P^S|}{|M_P^S| + |M_S^P|}$$

where the set of shared documents is defined as $O_P^S = P \cap S$ and $M_P^S = S \setminus O_P^S$ is the marginal set of documents classified by S and not classified by P (similarly $M_S^P = P \setminus O_P^S$). The following equivalence applies $O_P^S = O_S^P$. Notice that "O" stands for "overlapping" and "M" stands for "Marginal set".

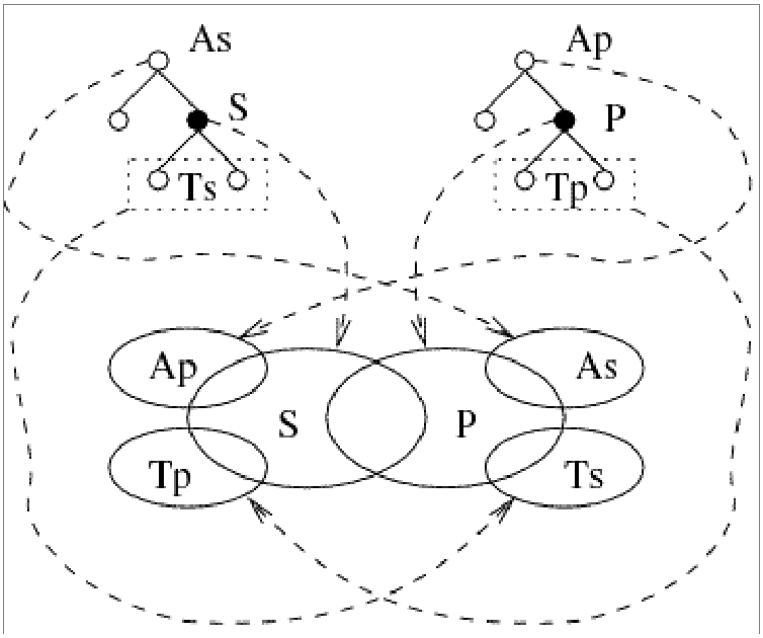

Figure 11.1: The pairwise relationships between two taxonomies.

We do a step forward because we do not only compute the equivalence hypothesis based on the notion of F1 measure of information retrieval, but we extend such equation to define the formulation of generalization and specialization hypotheses. Generalization and specialization hypotheses can be formulated taking advantage of the contextual encoding of knowledge in terms of hierarchies of categories. The challenge is to formulate a generalization hypothesis (and conversely a specialization hypothesis) between two nodes looking at the overlapping of set of documents classified in the ancestor or subtree of the reference nodes [Avesani, 2002].

The *generalization* relationship holds when the first node has to be considered more general than the second node. Intuitively, it happens when the documents classified under the first nodes occur in the ancestor of the second node, or the documents classified under the second node occur in the subtree of the first node. Following this intuition we can formalize the generalization

hypothesis as

$$Generalization = \frac{|O_P^S| + |O_{A_S}^P| + |O_{T_P}^S|}{|M_P^S| + |M_S^P|}$$

where $O_{A_S}^P$ represents the set of documents resulting from the intersection between M_S^P and the set of documents classified under the concepts in the hierarchy above S (i.e. the ancestors); similarly $O_{T_P}^S$ represents the set of documents resulting from the intersection between M_P^S and the set of documents classified under the concepts in the hierarchy below P (i.e. the children).

In a similar way we can design the *specialization* relationship. The first node is more specific than the second node when the meaning associated to the first node can be subsumed by the meaning of the second node. Intuitively, this happens when the documents classified under the first nodes occur in the subtree of the second node, or the documents classified under the second node occur in the ancestor of the first node.

$$Specialization = \frac{|O_P^S| + |O_{T_S}^P| + |O_{A_P}^S|}{|M_P^S| + |M_S^P|}$$

where $O_{T_S}^P$ represents the set of documents resulting from the intersection between M_S^P and the set of documents classified under the concepts in the hierarchy below S (i.e. the children); similarly $O_{A_P}^S$ represents the set of documents resulting from the intersection between M_P^S and the set of documents classified under the concepts in the hierarchy above P (i.e. the ancestors).

The three definitions above allow us to compute a relationship hypothesis between two nodes of two different taxonomies. Such an hypothesis relies on the assumption that if two nodes classify the same set of documents, the meaning associated to the nodes is reasonably the same. Of course this assumption is true for a virtually infinite set of documents. In a real world case study we face with finite set of documents, and therefore, this way of proceeding is prone to error. Nevertheless, our claim is that the approximation introduced by our assumption is balanced by the benefit of scaling with the annotation of large taxonomies.

11.1 Classification Hierarchies

Let us try to apply the notion of document-driven interpretation to a real world case study. We focus our attention to web directories for many reasons. Web directories are widely used and known; moreover they are homogeneous, that is they cover general topics. The meaning of a node in a web directory is not defined with formal semantics but by pragmatics. Furthermore the web directories address the same space of documents, therefore the working hypothesis of co-occurence of documents can be sustainable. Of course different web directories don't cover the same portion of the web but the overlapping is meaningful. The case study of web directories meets two requirements of the matching problem: to have heterogeneous representations of the same topics and to have taxonomies of large dimensions.

We address three main web directories: Google, Yahoo! and Looksmart. Nodes have been considered as categories denoted by the lexical labels, the tree structures have been considered as hierarchical relations, and the URL classified under a given node as documents. The following table summarizes the total amount of processed data.

Web Directories	Google	Looksmart	Yahoo!
number of nodes	335.902	884.406	321.585
number of urls	2.425.215	8.498.157	872.410

Let us briefly describe the process by which we have arranged an annotated corpus of pairwise relations between web directories.

Step 1. We crawled all three web directories, both the hierarchical structure and the web contents, then we computed the subset of URLs classified by all of them.

Step 2. We pruned the downloaded web directories by removing all the URLs that were not referred by all the three web directories.

Step 3. We performed an additional pruning by removing all the nodes with a number of URLs under a given threshold. In our case study we fixed such a threshold at 10.

Step 4. We manually recognized potential overlapping between two branches of two different web directories like

```
   Google:/Top/Science/Biology
Looksmart:/Top/Science-and-Health/Biology

    Yahoo:/Top/Computers-and-Internet/Internet
Looksmart:/Top/Computing/Internet

   Google:/Top/Reference/Education
    Yahoo:/Top/Education
```

We recognized 50 potential overlapping and for each of them we run an exhaustive assessment on all the possible pairs between the two related subtrees. Such an heuristic allowed us to reduce the quadratic explosion of cartesian product of two web directories. We focussed the analysis on smaller subtrees where the overlaps were more likely.

Step 5. We computed the three document-driven hypothesis for *equivalence*, *generalization* and *specialization* relationships as described above. Hypotheses of equivalence, generalization and specialization are normalized and estimated by a number in the range [0,1]. Since the cumulative hypothesis of all three relationships for the same pair of nodes can not be higher than 1, we introduce a threshold to select the winning hypothesis. We fixed such a threshold to 0.5.

We discarded all the pairs where none of the three relationship hypotheses was detected. This process allowed us to obtain 2265 pairwise relationships defined using the document-driven interpretation. Half are equivalence relationships and half are generalization relationships (notice that by definition generalization and specialization hypothesis are symmetric).

11.2 Thesauri and Ontologies

Validation of the results on the medical ontologies matching task is still an open problem. The results can be replicated in straightforward way. At the same time there are no sufficiently big set of the reference mappings what makes impossible calculation of the matching quality measures. In contrast to the generation of reference alignments for classification hierachies, we do not assume that instance data is available or that the models are represented in the same way or using the same language. Normally, the models will be from the same domain (eg. medicine or business). The methodology consists of four basic steps. In the first step, basic decisions are made about the representation of the conceptual models and instance data to be used. In the second step instance data is created by selecting it from an existing set or by classifying data according to the models under consideration. In the third step, the generated instance data is used to generate candidate mappings based on shared instances. In the forth step finally, the candidate mappings are evaluated against a set of quality criteria and the final set of reference mappings is determined.

Step 1. Preparation

The first step of the process is concerned with data preparation. In particular, we have to transform the conceptual models into a graph representation and select and prepare the appropriate instance data to be used to analyze overlap between concepts in the different models. We structure this step based on the KDD process for Knowledge Discovery and Data Mining.

Step 2. Instance Classification

In the second step the chosen instance data is classified according to the different conceptual models. For this purpose, an appropriate classification method has to be chosen that fits the data and the conceptual model. Further, the result of the classification process has to be evaluated. For this step we rely on established methods from Machine Learning and Data Mining.

Step 3. Hypothesis Generation

In the third step, we generate hypothesis for reference mappings based on shared instances created in the first two steps. In this step, we prune the classification by removing instances that are classified with a low confidence and selecting subsets of the conceptual models that show sufficient overlap. We further compute a degree of overlap between concepts in the different models and based on this degree of overlap select a set of reference mappings between concepts with a significant overlap.

Step 4. Evaluation

In the last step, the generated reference mapping is evaluated against the result of different matching systems as described in [Avesani *et al.*, 2005] using a number of criteria for a reference mapping. These criteria include correctness, complexity of the mapping problem and the ability of the mappings to discriminate between different matching approaches.

We are testing this methodology using a data set of medical documents called OHSUMED. The data set contains 350.000 articles from medical journals covering all aspects of medicine.

For classifying these documents according to the two ontologies of anatomy, we use the Collexis text indexing and retrieval system that implements a number of automatic methods for assigning concepts to documents. Currently, we are testing the data set and the system on a subset of UMLS with known mappings in order to assess the suitability of the methodology. The generation of the reference mappings for the Anatomy case will proceed around the end of 2005 and we are hopeful to have thoroughly tested set of reference mappings for the 2006 alignment challenge.

11.3 Evaluation Results

The evaluation was designed in order to assess the major dataset properties namely:

- *Complexity*, namely the fact that the dataset is "hard" for state of the art matching systems.
- *Discrimination ability*, namely the fact that the dataset can discriminate among various matching approaches.
- *Incrementality*, namely the fact that the dataset allows to incrementally discover the weaknesses of the tested systems.
- *Correctness*, namely the fact that the dataset can be a source of correct results.

We have evaluated two state of the art matching systems *COMA* and $S - Match$ and compared their results with *baseline solution*. Let us describe the matching systems in more detail.

The *COMA* system [Do and Rahm, 2001] is a generic syntactic schema matching tool. It exploits both element and structure level techniques and combines the results of their independent execution using several aggregation strategies. *COMA* provides an extensible library of matching algorithms and a framework for combining obtained results. Matching library contains 6 individual matchers, 5 hybrid matchers and 1 reuse-oriented matcher. One of the distinct features of the COMA tool is the possibility of performing iterations in the matching process. In the evaluation we used default combination of matchers and aggregation strategy (*NamePath+Leaves* and *Average* respectively).

S-Match is a generic semantic matching tool. It takes two tree-like structures and produces a set of mappings between their nodes. *S-Match* implements semantic matching algorithm in 4 steps. On the first step the labels of nodes are linguistically preprocessed and their meanings are obtained from the Oracle (in the current version WordNet 2.0 is used as an Oracle). On the second step the meaning of the nodes is refined with respect to the tree structure. On the third step the semantic relations between the labels at nodes and their meanings are computed by the library of element level semantic matchers. On the fourth step the matching results are produced by reduction of the node matching problem into propositional validity problem, which is efficiently solved by SAT solver or ad hoc algorithm (see [Giunchiglia *et al.*, 2004, Giunchiglia *et al.*, 2005] for more details).

We have compared the performance of these two systems with *baseline solution*. It is executed for each pair of nodes in two trees. The algorithm considers a simple string comparison among

Table 11.1: Evaluation Results

	Google vs. Looksmart	Google vs. Yahoo	Looksmart vs.Yahoo	Total
COMA	608	250	18	876 (38,68%)
$=$	608	250	18	876
$\subseteq$	N/A	N/A	N/A	N/A
$\supseteq$	N/A	N/A	N/A	N/A
S-Match	584	83	2	669 (29,54%)
$=$	2	5	0	7
$\subseteq$	46	19	2	67
$\supseteq$	536	59	0	595
Baseline	54	76	0	130 (5,39%)
$=$	52	0	0	52
$\subseteq$	0	76	0	76
$\supseteq$	2	0	0	2

the labels placed on the path spanning from a node to the root of the tree. Equivalence, more general and less general relations are computed as the corresponding logical operations on the sets of the labels.

The systems have been evaluated on the dataset described in Section 8.2.1. We computed the number of matching tasks solved by each matching system. Notice that the matching task was considered to be solved in the case when the matching system produce specification, generalization or equivalence semantic relation for it. For example, TaxME suggests that specification relation holds in the following example:

```
Google:/Top/Sports/Basketball/Professional/NBDL
Looksmart:/Top/Sports/Basketball
```

COMA produced for this matching task 0.58 similarity coefficient, which can be considered as equivalence relation with probability 0.58. In the evaluation we consider this case as true positive for *COMA* (i.e., the mapping was considered as found by the system).

Notice that at present TaxME contains only true positive mappings. This fact allows to obtain the correct results for Recall measure, which is defined as a ratio of reference mappings found by the system to the number of reference mappings. At the same time Precision, which is defined as ratio of reference mappings found by the system to the number of mappings in the result, can not be correctly estimated by the dataset since, as from Section 8.2.1, TaxME guarantee only the correctness but not completeness of the mappings it contains.

Evaluation results are presented on Table 11.1. It contains the total number of mappings found by the systems and the partitioning of the mappings on semantic relations. Let us discuss the results through the major dataset properties perspective.

11.3.1 Complexity

As from Table 11.1, the results of *baseline* are surprisingly low. It produced slightly more than 5% of mappings. This result is interesting since on the previously evaluated datasets (see [Bouquet *et al.*, 2003] for example) the similar baseline algorithm performed quite well and

found up to 70% of mappings. This lead us to conclusion that the dataset is not trivial (i.e., it is essentially hard for simple matching techniques).

As from Figure 11.2, *S-Match* found about 30% of the mappings in the biggest (Google-Yahoo) matching task. At the same time it produced slightly less than 30% of mappings in all the tasks. *COMA* found about 35% of mappings on Google-Looksmart and Yahoo-Looksmart matching tasks. At the same time it produced the best result on Google-Yahoo. *COMA* found slightly less than 40% of all the mappings. These results are interesting since, as from [Do and Rahm, 2001, Giunchiglia *et al.*, 2004], previously reported recall values for both systems were in 70-80% range. This fact turn us to conclusion that the dataset is hard for state of the art syntactic and semantic matching systems.

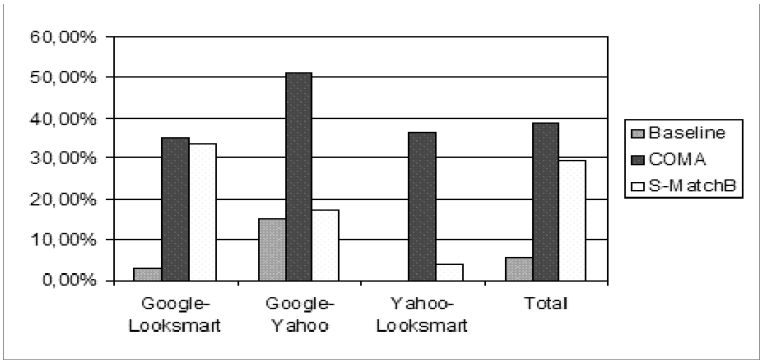

Figure 11.2: Percentage of correctly determined mappings(Recall)

11.3.2 Discrimination ability

Consider Figure 11.3. It presents the partitioning of the mappings found by *S-Match* and *COMA*. As from the figure the sets of mappings produced by *COMA* and *S-Match* intersects only on 15% of the mappings. This fact turns us to an important conclusion: the dataset is discriminating (i.e., it contains a number of features which are essentially hard for various classes of matching systems and allow to discriminate between the major qualities of the systems).

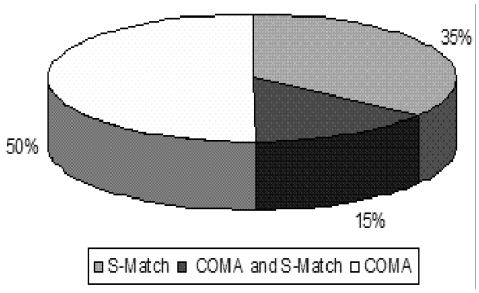

Figure 11.3: Partitioning of the mappings found by COMA and S-Match

11.3.3 Incrementality

In order to evaluate incrementality we have chosen *S-Match* as a test system. In order to identify the shortcomings of *S-Match* we manually analyzed the mappings missed by *S-Match*. This analysis allowed us to clasterize the mismatches into several categories. We describe in detail one of the most important categories of mismatches namely *Meaningless labels*.

Consider the following example:

```
Google:/Top/Science/Social_Sciences/Archaeology/Alternative/
       South_America/Nazca_Lines
Looksmart:/Top/Science_&_Health/Social_Science/Archaeology/
       By_Region/Andes_South_America/Nazca
```

In this matching task some labels are meaningful in the sense they define the context of the concept. In our example these are *Social_Sciences*, *Archaeology*, *South_America*, *Nazca*. The other labels do not have a great influence on the meaning of concept. At the same time they can prevent *S-Match* from producing the correct semantic relation. In our example *S-Match* can not find any semantic relation connecting *Nazca_Lines* and *Nazca*. The reason for this is *By_Region* label, which is meaningless in the sense it is defined only for readability and taxonomy partitioning purposes. An other example of this kind is

```
Google:/Top/Arts/Celebrities/A/Affleck,_Ben
Looksmart:/Top/Entertainment/Celebrities/Actors/Actors_A/
        Actors_Aa-Af/Affleck,_Ben/Fan_Dedications
```

Here, *A* and *Actors_A/Actors_Aa-Af* do not influence on the meaning of the concept. At the same time they prevent *S-Match* to produce the correct semantic relation holding between the concepts.

An optimized version of *S-Match* (*S-Match++*) has a list of meaningless labels. At the moment the list contains only about 30 words but it is automatically enriched in preprocessing phase. A general rule for considering natural language label as meaningless is to check whether it is used for taxonomy partitioning purposes. For example, *S-Match++* consider as meaningless the labels with the following structure *by* $\langle word \rangle$, where $\langle word \rangle$ stands for any word in natural language. However, this method is not effective in the case of labels composed from alphabet letters (such as *Actors_Aa-Af* from previous example). *S-Match++* deals with the latter case in the following way: the combination of letters are considered as meaningless if it is not recognized by WordNet, not in abbreviation or proper name list, and at the same time its length is less or equal to 3. The addition of these techniques allowed to improve significantly the *S-Match* matching capability. The number of mappings found by the system on TaxME dataset increased by 15%. This result gives us an evidence to incrementality of the dataset (i.e., the dataset allows to discover the weaknesses of the systems and gives the clues to the systems evolution).

Analysis of *S-Match* results on TaxME allowed to identify 10 major bottlenecks in the system implementation. At the moment we are developing ad hoc techniques allowing to improve *S-Match* results in this cases. The current version of *S-Match* (*S-Match++*) contains the

techniques allowing to solve 5 out of 10 major categories of mismatches. Consider Figure **??**.

11.3.4 Correctness

We manually analyzed correctness of the mappings provided by TaxME. At the moment 60% of mappings are processed and only 2-3% of them are not correct. Taking into account the notion of idiosyncratic classification [Goren-Bar and T.Kuflik, 2005] (or the fact that human annotators on the sufficiently big and complex dataset tend to have resemblance up to 20% in comparison with their own results), such a mismatch can be considered as marginal.

Chapter 12

Alternative tracks

We propose here a number of possible new tracks for next evaluation campaign. Their goal is to evaluate differently existing systems or to evaluate other features of the systems. One obvious feature that still has to be investigated is the time taken by algorithms.

12.1 Unconstrained discovery scenario

The currently used scenario of alignment evaluation is that of a contest. Its rules encourage the participants to seek maximal quality of individual alignments, which can be subsequently evaluated in terms of precision/recall with respect to a "golden standard" results (defined a priori but hidden from the participants). However, focussing on numerically measures of quality only is somewhat limiting wrt. the scope of observations potentially produced by automated alignment tools: more sophisticated observations, which can give interesting insight into the nature of tools as well as that of data but cannot be evaluated by traditional metrics, could arise. Let us consider, for example, alignments of contiguous paths in a tree, or "crossed alignments" that invert the taxonomic relationship. For ontologies with axioms (see the newly introduced notion of "Parallel OWL-DL ontologies"), even the logical difference in definitions could be explicitly captured. Sharing such heterogeneous observations clearly goes beyond the "contest" scenario of evaluation, as they can only be evaluated for subjective "interestingness".

There is an analogy with similar evaluation activities in the more traditional research area of Knowledge Discovery in Databases; discovery of ontological alignments can indeed be viewed as special case of knowledge discovery ("data mining"). The KDD Cup[1] enforces the type of analysis to be performed on the given dataset, compares the results obtained by different tools with correct results known a priori, yields a ranking, and, finally, awards the winner. On the other hand, in the Discovery Challenge [2], no correct results are known in advance: the researchers analyze the same data in different ways and with different methods, and share observations about their heterogeneous results (within a dedicated workshop).

The main advantage of this approach is the relatively unrestricted range of tasks to be carried out on data, and even the possibility to publish and discuss negative results, which may often be as

[1] http://kdd05.lac.uic.edu/kddcup.html
[2] http://lisp.vse.cz/challenge

useful as positive ones. A possible different track for future editions of the Ontology Alignment Evaluation Initiative could be an open workshop for different participants to contribute discussing and "negociating" the alignments. If some consensus is made, this can be further used as golden standards for other experiments.

12.2 Parallel OWL-DL ontologies

Within a recently launched informal initiative nicknamed OntoFarm, a new collection is being built by joint effort of multiple independent contributors from European research institutes (within as well as outside Knowledge Web). The chosen domain is that of conference organisation, including both programme and technical matters. To date, a pilot ontology (with about 50 concepts, 30 properties, and numerous axioms) derived from the structure of the EKAW conference) exist, and about 4-5 other are envisaged to arise in early Spring 2006. Due to its following characteristics, the new collection should improve on the tests cases provided in previous issues.

Richness in OWL-DL Constructs Most existing alignment tests focus on taxonomies of terms. However, many semantic web application scenarios assume complex ontologies that allow non-trivial reasoning. The design of the new collection will explicitly address the inclusion of full OWL-DL axiom types.

Larger size of collection Most existing alignment tests are limited to a pair of ontologies only. Here we consider multiple ontologies describing the same domain. This enables to consider more complex (meta-)alignment patterns, for example, such that some matching segments from two ontologies do not have match in the third one.

Interpretability by Non-Experts Despite Real-World flavor Complex real-world ontologies (such as those from the bio/medical domain) require a domain expert to properly interpret their concepts, while knowledge engineers can only handle them at the technical level. Here we intentionally chose a domain that is perfectly understandable (at least) to any researcher. On the other hand, by our experience with building the pilot ontology, the domain is non-trivial, shares many aspects with heavier-weighted industrial activities, and can give rise to numerous concepts, properties and axioms. Each ontology will model the domain of conference organization from the point of view of a concrete conference series its developer is deeply involved with. We thus believe that the collection, itself being "artificial" (i.e. created on purpose), will have heterogeneity introduced in a natural way, and its analysis will thus mimic real-life semantic web scenarios reasonably well.

Availability of Instance Data While in applications like business or medical applications, real instance data are subject to strong privacy constraints, data about organizers, committees, authors, presenters etc. of conferences are typically public and can even be picked-up from websites with reasonable effort. Information Extraction and Wrapper technology (also developed at UEP) can serve well here.

Instant Gratification for Ontology Development While the benefits of existing alignment tests were mostly cropped by the developers of tools, the new collection will aim at remunerating the developers of ontologies themselves. The collection will be equipped with a simple HTML-based front end giving access not only to the ontologies themselves (via a conventional query interface) but also to directly usable alignment and distributed reasoning results. Initial version of the HTML front end is described in [Svab *et al.*, 2005]. As the ontologies in the collection mirror the structure of real-world entities (namely, actual conference series), their alignment can give insights into this structure. Some candidate pay-off tasks, stimulating the development of further additions, are:

- Advertising the conference to the right target group, namely, offering the potential paper authors a conference that is thematically close, within reasonable time, in a certain (type of) location, with PC members from their institute/country and the like.
- Making the organizers aware of relationships (e.g., overlaps, personal links) with other conferences.
- Offering the conference organizers a suitable software tool that could support the organization of the event.

12.3 Thesaurus Alignment

The OAEI held in 2005 contained two different real world alignment tasks. One focussed on mapping Medical heavy-weight ontologies like OpenGalen and the other on mapping directory structures like Looksmart. The most widely used ontologies however are thesauri that lie in between these two kinds of ontologies in both richness and size. Thesauri are linguistic models that have been engineered to facilitate finding the right word to denote something. In libraries thesauri terms have been used to categorize publications ever since sorting books became necessary. Recently, thesauri have taken a leap in popularity, because the advent of the World Wide Web has made it easy for organizations to open their knowledge organization systems to the rest of the world. The Semantic Web community realized that semantic negotiation of such opened resources is necessary, which lead to the creation of the Simple Knowledge Organization System (SKOS[3].

SKOS SKOS consists of thee vocabularies: The SKOS Core Vocabulary, which contains the main classes and properties that are necessary to encode everything from controlled vocabularies to thesauri; the SKOS Mapping Vocabulary, which contains properties to create mappings between SKOS vocabularies; and SKOS Extensions, which contains domain-specific extensions to the SKOS Core and SKOS Mapping Vocabularies.

SKOS Collections Many organizations world-wide have started converting their thesauri to SKOS. Two such organizations are the Food and Agriculture Organization (FAO) of the United Nations and the United States National Agricultural Library (NAL). The FAO has converted their multilingual AGROVOC thesaurus[4], which consists of more than 16.000 terms into SKOS and is currently working on extending it with OWL statements. AGROVOC covers many subjects related to food and agriculture, such as fishing, famine and forestry. The NAL will release a SKOS version of their (monolingual english) NAL Agricultural Thesaurus in january 2006 [5]. The NAL

[3] `http://www.w3.org/2004/02/skos`, see Deliverable D2.2.6 [Euzenat *et al.*, 2005].
[4] `http://www.fao.org/agrovoc`
[5] `http://agclass.nal.usda.gov/agt`

thesaurus will consist of more than 41.000 terms covering an equally broad spectrum of subjects like food, agriculture, and the environment. Both thesauri are used to index large actively maintained research libraries, which are heavily used by researchers all over the world, such as food product developers, researchers investigating food-safety, and environmental policy makers.

Thesaurus Mapping Task Proposal A possible additional track for a further OAEI campaign could focus on creating a SKOS mapping between the SKOS versions of the AGROVOC and NAL thesauri. The mapping task is suitable for the OAEI, because of the following reasons:

- Both thesauri are large.
- They are widely used.
- The thesauri cover much of the same subjects.
- The concepts covered in the thesauri are understandable to people that are not a domain expert (For example, semantic web researchers.)
- The SKOS Mapping Vocabulary is an important, applied (would be) standard.
- Many of the corpora indexed with terms from the thesauri (instance data) are freely accessible on the web.

It however suffers from the following shortcommings:

- It is useful for thesauri aligners rather than ontology aligner.
- Delivering in SKOS is not prone to be integrated into the current evaluation platform (while delivering in the alignment API, will before the next evaluation provide SKOS generation).
- As for the anatomy example, there is currently no accepted mapping between these ontologies so the evaluation problem remain the same as this year.

12.4 Full real-world problem solving

One observation that was made is that we have trouble evaluating "real world" test cases. Moreover, by evaluating features of alignment, we do not evaluate their value in context, i.e., for solving real problems. In context, it can be possible to compare the performance of the systems without knowing an absolutely true correct alignment. We could measure if the system performs better as a whole in a (semi-)operational context.

For that purpose, we would like to have proposal challenge from real users who need ontology alignment. The ontology would be provided by the use case provider as well as the success criterion and some infrastructure for pluging alignments to be used. The organisers could provide help for setting the evaluation protocol.

This would help us having an independently submitted and independently evaluated real-world problem to solve; this would help the submitter having help from the community as a whole to solve her problem. Moreover, it is more gratifying for participant to know that they have contributed improving the solution to some real-world problem.

Part IV

Conclusions

The work reported in this deliverable shows that KnowledgeWeb has succeeded in setting up an alignment challenge that attracts attention not only inside the network, which is demonstrated by the participation of research institutes from the North America and Asia in the two campaigns so far. The alignment challenge follows a clear methodology that has been described in detail in deliverable 2.2.3 and refined in this deliverable and has lead to advances in the state of the art in ontology alignment techniques which is demonstrated in part 1 of this deliverable.

A major aspect of this deliverable is to show how both the design of the evaluation methodology as well as the methods competing in the challenge evolve. The tests that have been run this year are harder and more complete than those of last year. However, more teams participated and the results tend to be better. This shows that, as expected, the field of ontology alignment is getting stronger (and we hope that evaluation is contributing to this progress). Reading the papers of the participants should help people involved in ontology matching to find what make these algorithms work and what could be improved.

Another noteworthy feature is that most of the tools that have been presented here are available on the web and many of them open source. This contributes independent scrutiny and improvement of the field as a whole.

The Ontology Alignment Evaluation Initiative has been created to carry this evaluation effort outside Knowledge web. It will continue these tests by improving both test cases and test methodology for being more accurate. It can be found at:

`http://oaei.inrialpes.fr`.

A number of obstacles to a successful evaluation of ontology alignment technology have been identified in the context of the evaluation. For example, problems related to quality measures for alignments are discussed in section 10 and proposals for measures that are better suited for the purpose than standard measures from information retrieval are made.

The most severe obstacle to a successful evaluation that has been identified is the lack of suitable test data. The reason for this is that there is a conflict between the need to measure the quality of the generated alignments and the wish to have realistic alignment problems. At the current stage, the data sets that support evaluation in terms of a standard alignment automatically generated ones can be compared are either rather small like the Benchmark data set in the challenge or rather inexpressive like the Directory data set in the challenge. Other data sets that are both, large and complex like the Anatomy data set in this years challenge are very hard to evaluate as it is entirely unclear how a correct mapping looks like. Even the Directory data set could only be evaluated for completeness, but not for correctness. We try to address this problem by suggesting a number of alternative data sets for future challenges, but the problem described above seems to be a fundamental dilemma.

A way to overcome this dilemma is to judge alignments not in terms of precision and recall but in terms of the usefulness of the generated mappings with respect to a concrete application.

The future plans for the Ontology Alignement Evaluation Initiative and Knowledge web work

package 2.2 are certainly to go ahead and improving the functioning of these evaluation campaign. This most surely involves:

- Finding new real world cases;
- Improving the tests along the lesson learned;
- Accepting continuous submissions (through validation of the results);
- Improving the measures to go beyond precision and recall.

Acknowledgements

We warmly thank each participant of this campaign. We know that they worked hard for having their results ready and they provided insightful papers presenting their experience.

Many thanks are due to the teams at the University of Washington and the University of Manchester for allowing us to use their ontologies of anatomy.

The members of the Ontology Alignment Evaluation Initiative Steering committee:

- Benjamin Ashpole (Lockheed Martin Advanced Technology Lab.),
- Marc Ehrig (University of Karlsruhe),
- Jérôme Euzenat (INRIA Rhône-Alpes),
- Lewis Hart (Applied Minds),
- Todd Hughes (Lockheed Martin Advanced Technology Labs),
- Natasha Noy (Stanford University),
- Heiner Stuckenschmidt (University of Mannheim),
- Petko Valtchev (Université de Montréal, DIRO)

Bibliography

[Avesani *et al.*, 2005] Paolo Avesani, Fausto Giunchiglia, and Michael Yatskevich. A large scale taxonomy mapping evaluation. In *Proceedings of International Semantic Web Conference (ISWC)*, 2005.

[Avesani, 2002] P. Avesani. Evaluation framework for local ontologies interoperability. In *AAAI Workshop on Meaning Negotiation*, 2002.

[Baeza-Yates and Ribeiro-Neto, 1999] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[Bechhofer *et al.*, 2003] Sean Bechhofer, Rapahel Voltz, and Phillip Lord. Cooking the semantic web with the OWL API. In *Proc. 2nd International Semantic Web Conference (ISWC), Sanibel Island (FL US)*, 2003.

[Bisson, 1992] Gilles Bisson. Learning in FOL with similarity measure. In *Proc. 10th American Association for Artificial Intelligence conference, San-Jose (CA US)*, pages 82–87, 1992.

[Bouquet *et al.*, 2003] P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. In Fensel D., Sycara K. P., and Mylopoulos J., editors, *2nd international semantic web conference (ISWC 2003)*, volume 2870 of *LNCS*, Sanibel Island, Fla., 20-23 October 2003.

[Cohen *et al.*, 2003] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*, pages 73–78, 2003.

[de Bruijn and Feier., 2005] J. de Bruijn and C. Feier. Report on ontology mediation for case studies. deliverable D4.6.1, SEKT, June 2005.

[Didion, 2004] John Didion. The java wordnet library, 2004. http://jwordnet.sourceforge.net/.

[Do and Rahm, 2001] H.H. Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proceedings of Very Large Data Bases Conference (VLDB)*, pages 610–621, 2001.

[Do *et al.*, 2002] Hong-Hai Do, Sergey Melnik, and Erhard Rahm. Comparison of schema matching evaluations. In *Proc. GI-Workshop "Web and Databases", Erfurt (DE)*, 2002. http://dol.uni-leipzig.de/pub/2002-28.

[Doan *et al.*, 2003a] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12(4):303–319, 2003.

[Doan *et al.*, 2003b] H. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning*, 50:279–301, 2003.

[Ehrig and Euzenat, 2005] Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In Ben Ashpole, Jérôme Euzenat, Marc Ehrig, and Heiner Stuckenschmidt, editors, *Proc. K-Cap 2005 workshop on Integrating ontology, Banff (CA)*, pages 25–32, 2005.

[Ehrig and Sure, 2004] M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In Christoph Bussler, John Davis, Dieter Fensel, and Rudi Studer, editors, *Proceedings of the First European Semantic Web Symposium*, volume 3053 of *Lecture Notes in Computer Science*, pages 76–91, Heraklion, Greece, MAY 2004. Springer Verlag.

[Ehrig and Sure, 2005] M. Ehrig and Y. Sure. Adaptive semantic integration. In *Proceedings of the ODBIS workshop at the 31st VLDB Conference*, Trondheim, Norway, September 2005.

[Ehrig *et al.*, 2003] M. Ehrig, P. Haase, F. van Harmelen, R. Siebes, S. Staab, H. Stuckenschmidt, R. Studer, and C. Tempich. The SWAP data and metadata model for semantics-based peer-to-peer systems. In *Proceedings of the First German Conference on Multiagent Technologies (MATES-2003)*, Lecture Notes in Artificial Intelligence. Springer, September 2003.

[Ehrig *et al.*, 2005] M. Ehrig, S. Staab, and Y. Sure. Bootstrapping ontology alignment methods with APFEL. In Y. Gil, E. Motta, and V.R. Benjamins, editors, *Proceedings of the Fourth International Semantic Web Conference (ISWC-2005)*, Lecture Notes in Computer Science, 2005.

[Euzenat and Valtchev, 2003] Jérôme Euzenat and Petko Valtchev. An integrative proximity measure for ontology alignment. In *Proc. ISWC-2003 workshop on semantic information integration, Sanibel Island (FL US)*, pages 33–38, 2003.

[Euzenat and Valtchev, 2004] Jérôme Euzenat and Petko Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI), Valencia (ES)*, pages 333–337, 2004.

[Euzenat *et al.*, 2003] Jérôme Euzenat, Nabil Layaïda, and Victor Dias. A semantic framework for multimedia document adaptation. In *Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco (MX)*, pages 31–36, 2003.

[Euzenat *et al.*, 2004] Jérôme Euzenat, Thanh Le Bach, Jesús Barrasa, Paolo Bouquet, Jan De Bo, Rose Dieng-Kuntz, Marc Ehrig, Manfred Hauswirth, Mustafa Jarrar, Rubén Lara, Diana Maynard, Amedeo Napoli, Giorgos Stamou, Heiner Stuckenschmidt, Pavel Shvaiko, Sergio Tessaris, Sven Van Acker, and Ilya Zaihrayeu. State of the art on ontology alignment. deliverable D2.2.3, Knowledge web NoE, 2004.

[Euzenat *et al.*, 2005] Jérôme Euzenat, François Scharffe, and Luciano Serafini. Specification of the delivery alignment format. deliverable 2.2.6, Knowledge web NoE, 2005.

[Euzenat, 2004] Jérôme Euzenat. An API for ontology alignment. In *Proc. 3rd international semantic web conference, Hiroshima (JP)*, pages 698–712, 2004.

[Fagin *et al.*, 2003] R. Fagin, P.G. Kolaitis, R.J. Miler, and L. Popa. Data exchange: Semantics and query answering. In *Proceedings of the 9th International Conference on Database Theory (ICDT'03)*, pages 207–224, Sienna, Italy, 2003.

[Freksa, 1992] Christian Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1–2):199–227, 1992.

[Gale and Shapley, 1962] David Gale and Lloyd Stowell Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 1962.

[Giunchiglia *et al.*, 2004] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In Bussler C., Davies J., Fensel D., and Studer R., editors, *1st european semantic web symposium (ESWS'04)*, volume 3053 of *LNCS*, Heraklion, 10-12 May 2004.

[Giunchiglia *et al.*, 2005] F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient semantic matching. In *Proceedings of the 2nd european semantic web conference (ESWC'05)*, Heraklion, 29 May-1 June 2005.

[Goren-Bar and T.Kuflik, 2005] D. Goren-Bar and T.Kuflik. Supporting user-subjective categorization with self-organizing maps and learning vector quantization. *Journal of the American Society for Information Science and Technology JASIST*, 56(4):345–355, 2005.

[Hamacher *et al.*, 1978] H. Hamacher, H. Leberling, and H.-J. Zimmermann. Sensitivity analysis in fuzzy linear programming. *Fuzzy Sets and Systems*, 1:269–281, 1978.

[Heß and Kushmerick, 2004] Andreas Heß and Nicholas Kushmerick. Iterative ensemble classification for relational data: A case study of semantic web services. In *Proceedings of the 15th European Conference on Machine Learning*, Pisa, Italy, 2004.

[Horrocks *et al.*, 2003] I. Horrocks, P.F. Patel-Schneider, and F. van Harmelen. From shiq and rdf to owl: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[Hu *et al.*, 2005] W. Hu, N. Jian, Y. Qu, and Y. Wang. Gmo: A graph matching for ontologies. In *Proceedings of the K-Cap 2005 Workshop on Integrating Ontologies*, pages 43–50, 2005.

[Ichise *et al.*, 2003] R. Ichise, H. Takeda, and S. Honiden. Integrating multiple internet directories by instance-based learning. In *IJCAI*, pages 22–30, 2003.

[Jian *et al.*, 2005] N. Jian, W. Hu, G. Cheng, and Y. Qu. Falcon-ao: Aligning ontologies with falcon. In *Proceedings of the K-Cap 2005 Workshop on Integrating Ontologies*, pages 87–93, 2005.

[Langlais *et al.*, 1998] Philippe Langlais, Jean Véronis, and Michel Simard. Methods and practical issues in evaluating alignment techniques. In *Proc. 17th international conference on Computational linguistics, Montréal (CA)*, pages 711–717, 1998.

[Levenshtein, 1966] I. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 1966.

[Melnik *et al.*, 2002] S. Melnik, H. Molina-Garcia, and E. Rahm. Similariy flooding: A versatile graph matching algorithm. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2002.

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

[Munkres, 1957] James Munkres. Algorithms for the assignment and transportation problems. *SIAP*, 5(1):32–38, 1957.

[Noy and Musen, 2003] N. F. Noy and M. A. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

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

[Qu *et al.*, 2005] Y. Qu, W. Hu, and G. Cheng. Constructing virtual documents for ontology matching. Submitted to WWW 2006, 2005.

[Salton, 1989] Gerard Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[Sebastiani, 2002] F. Sebastiani. Machine learning in automated text categorization. *ACM Comuting Surveys*, 34(1):1–47, 2002.

[Stoilos *et al.*, 2005] G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. In Y. Gil, editor, *Proceedings of the International Semantic Web Conference(ISWC)*, volume 3729 of *Lecture Notes in Computer Science*, pages 624–637. Springer-Verlag, 2005.

[Straccia and Troncy, 2005a] Umberto Straccia and Raphaël Troncy. oMAP: An implemented framework for automatically aligning owl ontologies. In *Proceedings of the 2^{nd} Italian Semantic Web Workshop (SWAP'05)*, Trento, Italy, 2005.

[Straccia and Troncy, 2005b] Umberto Straccia and Raphaël Troncy. oMAP: Combining classifiers for aligning automatically owl ontologies. In *Proceedings of the 6^{th} International Conference on Web Information Systems Engineering (WISE'05)*, pages 133–147, New York City, New York, USA, 2005.

[Straccia and Troncy, 2005c] Umberto Straccia and Raphaël Troncy. oMAP: Results of the ontology alignment contest. In *Proceedings of the K-Cap 2005 Workshop on Integrating Ontologies*, pages 92–96, 2005.

[Sun and Lin, 2001] Aixin Sun and Ee-Peng Lin. Hierarchical text classification and evaluation. In *Proc. IEEE international conference on data mining*, pages 521–528, 2001.

[Sure *et al.*, 2004] York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the 3rd International Workshop on Evaluation of Ontology based Tools (EON)*, Hiroshima, Japan, 2004.

[Svab *et al.*, 2005] O. Svab, V. Svatek, P. Berka, D. Rak, and P. Tomasek. Ontofarm: Towards an experimental collection of parallel ontologies. In *Proceedings of the 5th International Semantic Web Conference ISWC-05*, 2005. Poster Track.

[Tounazi, 2004] Mohamed Tounazi. Alignement d'ontologies dans OWL. Master's thesis, University of Montréal, 2004.

[Valtchev, 1999] Petko Valtchev. *Construction automatique de taxonomies pour l'aide à la représentation de connaissances par objets*. Thèse d'informatique, Université Grenoble 1, 1999.

[van Rijsbergen, 1979] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[Winkler, 1999] William Winkler. The state record linkage and current research problems. Technical report, Statistics of Income Division, Internal Revenue Service Publication, 1999.

Related deliverables

A number of Knowledge web deliverable are clearly related to this one:

Project	Number	**Title** and relationship
KW	D2.1.1	**Survey of scalability techniques for reasoning with ontologies** provided an in-depth discussion about benchmarking techniques that have been mentioned here.
KW	D2.1.4	**Specification of a methodology, general criteria, and test suites for benchmarking ontology tools** provides a framework along which to define a benchmarking test.
KW	D2.2.1	**Specification of a common framework for characterizing alignment** provided the framework for us to define the benchmarking actions.
KW	D2.2.3	**State of the art on ontology alignment** provides a panorama of many of the techniques that must be evaluated in the current deliverable.