# Specification of a benchmarking methodology for alignment techniques

Jérôme Euzenat, Raúl García Castro, Marc Ehrig


# D2.2.2: Specification of a benchmarking methodology for alignment techniques

**Coordinator: Jérôme Euzenat (INRIA Rhône-Alpes)**
**Raúl García Castro (UP Madrid),**
**Marc Ehrig (Universität Karlruhe),**

**Abstract.**
This document considers potential strategies for evaluating ontology alignment algorithms. It identifies various goals for such an evaluation. In the context of the Knowledge web network of excellence, the most important objective is the improvement of existing methods. We examine general evaluation strategies as well as efforts that have already been undergone in the specific field of ontology alignment. We then put forward some methodological and practical guidelines for running such an evaluation.
Keyword list: ontology matching, ontology alignment, ontology mapping, evaluation, benchmarking, contest, performance measure.



| | |
|---|---|
| Document Identifier | KWEB/2004/D2.2.2/v1.0 |
| Project | KWEB EU-IST-2004-507482 |
| Version | v1.0 |
| Date | February 2, 2005 |
| State | final |
| 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 Karlsruhe
University of Manchester
University of Sheffield
University of Trento
Vrije Universiteit Amsterdam
Vrije Universiteit Brussel

# Changes

| Version | Date | Author | Changes |
|---|---|---|---|
| 0.1 | 12.07.2004 | Jérôme Euzenat | creation |
| 0.2 | 29.09.2004 | Jérôme Euzenat | improving outline |
| 0.3 | 17.11.2004 | Jérôme Euzenat | filled chapter 1, 4 and appendice |
| 0.4 | 26.11.2004 | Jérôme Euzenat | filled measures; reorganised |
| 0.5 | 15.12.2004 | Jérôme Euzenat/Marc Ehrig | updated measures/introductioncompleted |
| 0.6 | 30.12.2004 | Jérôme Euzenat/Raul Garcia Castro | Summary/exp. dimensions/setup/rules |
| 0.7 | 07.01.2005 | Jérôme Euzenat | Revision (submitted to quality control) |
| 1.0 | 31.01.2005 | Jérôme Euzenat | Revision (implementation of quality control comments) |

# Executive Summary

Heterogeneity problems on the semantic web can be solved, for some of them, by aligning heterogeneous ontologies. Aligning ontologies consists of providing the corresponding entities in these ontologies. This process is precisely defined in deliverable 2.2.1. Actual algorithms have been presented in deliverable 2.2.3 together with a number of use cases of ontology alignment. Many techniques are available for achieving ontology alignment and many systems have been developed based on these techniques. However, few comparisons and few integration is actually provided by these implementations.

The present deliverable studies what kind of evaluation can be carried out on alignment algorithms in order to help the worldwide research community to improve on the current techniques.

In the current document, we first examine the purpose and types of evaluation as well as established evaluation methodology (§1). We found that two different kinds of benchmarks are worth developing for ontology alignment: competence benchmarks based on many "unit tests" which characterise a particular situation and enable to assess the capabilities of each algorithms and performance benchmarks based on challenging "real-world" situations in which algorithms are in competition.

We have examined the possible variations of the ontology alignment problem (§2) and the possible measures that can be used for evaluating alignment results (§3). This allows us to specify the profile of the kind of benchmarks to be performed and how results will be evaluated. The variation opportunities are very large so we had to restrict the considered task (at least for competence benchmarks) drastically. These restrictions could be relaxed in further evaluation or when considering and evaluating algorithms on a particular, clearly identified subtask. Concerning the evaluation measure, precision and recall are, so far, the best understood measures. However, it will be very important in the future to involve resource consumption measures.

Then we draw on previous experiments in order to design some guidelines for performing an evaluation campaign. This involves defining a set of rules for the evaluation (§4) based on a committee and other evaluation initiatives. We also define guidelines for setting up the evaluation material (§5). This depends on the type of evaluation to carry out. Competence benchmarks can be evaluated against a semi-automatically generated set of tests around a reference ontology. Performance benchmarks require the identification of relevant ontology and suitable reference alignment. We provides a number of candidate for further tests. Last, we describe valuable tools for automating the evaluation process (§6): generating tests, running tests and evaluating results. These tools are in part already implemented.

An appendix presents two evaluation initiatives to which we participated in 2004.

# Contents

# Chapter 1

# Introduction: purpose, method and types of evaluation for ontology alignment

Aligning ontologies consists of finding the corresponding entities in these ontologies. There have been many different techniques proposed for implementing this process (see deliverable 2.2.3). They can be classified along the many features that can be found in ontologies (labels, structures, instances, semantics), or with regard to the kind of disciplines they belong to (e.g., statistics, combinatorics, semantics, linguistics, machine learning, or data analysis) [Rahm and Bernstein, 2001; Kalfoglou and Schorlemmer, 2003; Euzenat *et al.*, 2004]. The alignment itself is obtained by combining these techniques towards a particular goal (obtaining an alignment with particular features, optimising some criterion). Several combination techniques are also used. The increasing number of methods available for schema matching/ontology integration suggests the need to establish a consensus for evaluation of these methods.

Beside this apparent heterogeneity, it seems sensible to characterise an alignment as a set of pairs expressing the correspondences between two ontologies. We proposed, in deliverable 2.2.1, to characterise an alignment as a set of pair of entities ($e$ and $e'$), coming from each ontologies ($o$ and $o'$), related by a particular relation ($R$). To this, many algorithms add some confidence measure ($n$) in the fact the relation holds [Euzenat, 2003; Bouquet *et al.*, 2004; Euzenat, 2004].

From this characterisation it is possible to ask any alignment method, given

- two ontologies to be aligned;
- an input partial alignment (possibly empty);
- a characterization of the wanted alignment (1:+, ?:?, etc.).

to output an alignment. From this output, the quality of the alignment process could be assessed with the help of some measurement. However, very few experimental comparison of algorithms are available. It is thus one of the objectives of Knowledge web and other people worldwide to run such an evaluation. We have participated in the organisation of two events in 2004 which are the premises of a larger evaluation event:

- The Information Interpretation and Integration Conference (I3CON), held at the NIST Performance Metrics for Intelligent Systems (PerMIS) Workshop, is an ontology alignment

demonstration competition on the model of the NIST Text Retrieval Conference. This contest has focused "real-life" test cases and comparison of algorithm global performance.
- The Ontology Alignment Contest at the 3rd Evaluation of Ontology-based Tools (EON) Workshop, held at the International Semantic Web Conference (ISWC), targeted the characterisation of alignment methods with regard to particular ontology features. This contest defined a proper set of benchmark tests for assessing feature-related behavior.

These two events are described more thoroughly in Appendix A.

Since all benchmarking activity must be carried out with a systematic procedure on clearly defined tasks. This is the purpose of this deliverable to propose such a procedure. This introduction will define the general objective of evaluating the alignment algorithms, the kind of tests which can be performed and the overall methodology to be followed.

## 1.1 Goal of evaluation

As mentionned in deliverable D2.1.1, evaluation should enable the measure of the degree of achievement of proposed tasks on a scale common to all methods. The main features of benchmarking are:

- measurement via comparison;
- continuous improvement;
- systematic procedure.

A benchmark is a test that measures the performances of a system or subsystem on a well defined task or set of tasks (comp.benchmark.FAQ). In fact, the two first items are not really the same goal and we will identify different types of evaluation later on.

The major and long term purpose of the evaluation of ontology alignment methods is to help designers and developers of such methods to improve them and to help users to evaluate the suitability of proposed methods to their needs. The benchmarking considered here should help research on ontology alignment. For that purpose, the evaluation should help evaluating absolute performances (e.g., compliance) and relative performances (e.g., in speed or accuracy).

The medium term goal is to set up a set of reference benchmark tests for assessing the strengths and weaknesses of the available tools and to compare them. Some of these tests are focussing the characterisation of the behaviour of the tools rather than having them compete on real-life problems. It is expected that they could be improved and adopted by the algorithm implementers in order to situate their algorithms. Building benchmark suites is highly valuable not just for the group of people that participates in the contests, but for all the research community. The evaluation should thus be run over several years in order to allow the measure of the evolution of the field.

The shorter term goal of the initiatives launched in 2004 was firstly to illustrate how it is possible to evaluate ontology alignment tools and to show that it was possible to build such an evaluation campaign. It is a common subgoal of evaluation campaign that their return helps improving the evaluation methodologies.

## 1.2 Types of evaluations

There can be several classifications of benchmarks depending on the criteria used. We can divide benchmarking with regard to what they are supposed to evaluate:

**competence benchmarks** allows to characterise the level of competence and performance of a particular system with regard to a set of well defined tasks. Usually, tasks are designed to isolate particular characteristics. This kind of benchmarking is relevant to kernel benchmark or unit tests;

**comparison benchmark** allows to compare the performance of various systems on a clearly defined task or application.

The goal of these two kinds of benchmarks are different: competence benchmarks aim at helping system designers to evaluate their systems and to localise them which regard with a common stable framework. It is helpful for improving individual systems. The comparison benchmarks enables to compare systems with regard to each others on a general purpose tasks. Its goal is mainly to help improving the field as a whole rather than individual systems. These two kinds of benchmarks are futher considered below.

In deliverable D2.1.4, the following classification, due to [Stefani *et al.*, 2003], describes the four following types of benchmarks that can be used in the evaluation of software systems:

**Application benchmarks** These benchmarks use real applications and workload conditions.

**Synthetic benchmarks** These benchmarks emulate the functionalities of significant applications, while cutting out additional or less important features.

**Kernel benchmarks** These benchmarks use simple functions designed to represent key portions of real applications.

**Technology-specific benchmarks** These benchmarks are designed to point out the main differences of devices belonging to the same technological family.

Each of these approaches have advantages and drawbacks. We will see that the I3CON experiment choose the first approach and ended with the second, while the EON initiative has used the fourth option.

This classification is concerned by the way to design benchmarks while the competence /performance classification is based on what is evaluated by the benchmarks. These two are not totally independent as the phrasing suggests it. Since we are first interested by the "what to evaluate" rather than the "how", we will focus on competence/performance.

### 1.2.1 Competence benchmark

Competence benchmarks aim at characterising the kind of task each method is good at. There are many different areas in which methods can be evaluated. One of them is the kind of features they use for finding matching entities (this complements the taxonomy provided in [Rahm and Bernstein, 2001]):

**terminological (T)** comparing the labels of the entities trying to find those which have similar names;

**internal structure comparison (I)** comparing the internal structure of entities (e.g., the value range or cardinality of their attributes);

**external structure comparison (S)** comparing the relations of the entities with other entities;

**extensional comparison (E)** comparing the known extension of entities, i.e. the set of other entities that are attached to them (in general instances of classes);

**semantic comparison (M)** comparing the interpretations (or more exactly the models satisfying the entities).

A set of reference benchmarks, targetting one type of feature at a time can be defined. These benchmarks would caracterize the competence of the method for one of these particular features of the languages.

### 1.2.2 Performance benchmarks: competition

Performance benchmarks are aimed at evaluating the overall behaviour of alignment methods in versatile real-life examples. It can be organised as a yearly or bi-annual challenge (à la TREC) for comparing the best compound methods. Such benchmarks should yield as a result the distance between provided output and expected result as well as traditional measures of the amount of resource consumed (time, memory, user input, etc.).

## 1.3 Evaluation methodology

Each evaluation must be carried out according to some methodology. Knowledge web deliverable D2.1.4 presents a benchmarking methodology that is briefly summarized here.

The benchmarking process defined in the methodology is a continuous process that should be performed indefinitely in order to obtain a continuous improvement both in the tools and in the same benchmarking process. This process is composed of a benchmarking iteration that is repeated forever and that is composed of three phases (Plan, Experiment, and Improve) and ends with a Recalibration task.

The three phases of each iteration are the following:

**Plan phase** It is composed of the set of tasks that must be performed for indentifying the goal and the subject of the evaluation (see above), preparing the proposal for benchmarking, finding other organisations that want to participate in the benchmarking activity (Section 2.2), and planning the benchmarking.

**Experiment phase** It is composed of the set of tasks where the experimentation over the different tools that are considered in the benchmarking activity is performed. This includes defining the experiment and its tool set, processing and analysing the data obtained, and reporting the experimentation results.

**Improve phase** It is composed of the set of tasks where the results of the benchmarking process are produced and communicated to the benchmarking partners, and the improvement of the different tools is performed in several improvement cycles. Precising how to report and communicate on the results is considered in §4; while planning the corrective methods, improving the actual systems and monitoring the results is a matter concerning algorithms developers and is not covered in this deliverable.

While the three phases mentioned before are devoted to the tool improvement, the goal of the **Recalibration** task is to improve the benchmarking process itself after each benchmarking iteration, using the lessons learnt while performing the benchmarking.

We are, in this deliverable, mainly concerned with the design of the evalutation. I.e., the Plan and Experiment phases described above. The processing and recalibrating of the evaluation is, in

theory the topic of deliverable D2.2.4. However, since, we already run two evaluation events in 2004, we have implemented these two steps already and this deliverable can be seen as the end of the recalibrating phase. The two experiments are reported in appendix. The remainder of this deliverable will consist in proposing mainly the main outline for the Plan phase. The Experiment phase will be the subject of deliverable 2.2.4.

## 1.4 Conclusion

The goal of the Knowledge web evalution effort is the improvement of ontology e techniques. For that purpose we will define the kind of tests to be processed and measures for assessing the results. This will be done for two kinds of tests: competence and performance benchmarks.

Next chapter evaluates what is the variability in the alignment task, and, consequently, what are the parameters that must be controlled in its evaluation. Chapter 3 considers the potential evaluation metrics that can be used in order to assess the performance of the evaluated algorithms. Chapter 4 provides the definition of a possible evaluation process, including the identification of actors and Chapter 6 describes the kind of support which is provided to the community in order to perform these evaluations. The last chapter will provide some guidelines for defining benchmark tests in order to evaluate alignments.

# Chapter 2

# Dimensions and variability of alignment evaluation

The goal of this chapter is to characterize the variability of the alignment task in order to assess the limitations of the benchmark tests or to design benchmarks spanning the whole spectrum of alignment and to know what variable must be controlled during their design.

Deliverable 2.2.1 provided a precise definition of the alignment process which is recalled here. The alignment process simply consists of generating an alignment ($A'$) from a pair of ontologies ($o$ and $o'$). However, there are various other parameters which can extend the definition of the alignment process. These are namely, the use of an input alignment ($A$) which is to be completed by the process, the alignment methods parameters (which can be weigths for instance) and some external resources used by the alignment process (which can be general-purpose resources not made for the case under consideration, e.g., lexicons, databases). This process can be defined as follow:

**Definition 1 (Alignment process).** *The alignment process can be seen as a function $f$ which, from a pair of ontologies $o$ and $o'$ to align, an input alignment $A$, a set of parameters $p$, a set oracles and resources $r$, returns a new alignment $A'$ between these ontologies:*

$$A' = f(o, o', A, p, r)$$

This can be represented as in Figure 2.1.

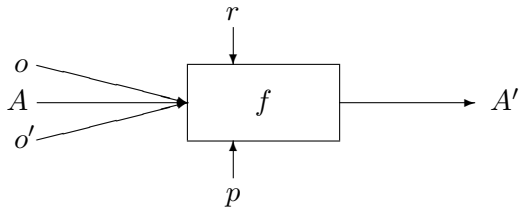


Figure 2.1: The alignment process.

Each of the elements featured in this definition can have specific characteristics which influence the difficulty of the alignment task. It is thus necessary to know and control these characteristics (called dimensions because they define a space of possible tests). The purpose of the

dimensions is the definition of the parameters and characteristics of expected behavior in benchmark. Indeed, for each dimension a specific benchmark could be designed. However, there are too many of them and it is thus necessary to choose fixed values for most of these possible parameters.

We review below all the dimensions and justify some choices in designing benchmarks.

## 2.1 Input ontologies

Input ontologies ($o$, $o'$) can be characterised by three different dimensions:

**Heterogeneity** of the input languages: are they described in the same knowledge representation languages? This corresponds to asking for the non emptyness of the syntactic component of the resulting alignment.

**Languages:** what are the languages of the ontologies? Example of languages are KIF, OWL, RDFS, UML, F-Logic, etc. as well as variant of these e.g., OWL-Lite, OWL-DL, OWL-Full.

**Number:** is this an alignment or a multi-alignment?

Currently, Knowledge web considers the alignment of ontologies expressed in the same language. The rationale for this is that language translation or language mapping resort to very specific techniques different from those used for aligning two ontologies. These techniques can be set up independently of any ontology. We thus consider that when confronted with ontologies expressed in different languages, it is better to first translate one of the ontology into the language of the other before processing an alignment properly speaking.

All the languages mentioned above are worth considering. However, in the setting up of a particular test, it is necessary to decide for the use of one language. During the first meetings of the 2.2 work package, it has been considered that the OWL language was the choice to consider first. Moreover, we decided for the OWL-DL fragment of OWL. During the first campaign we run, some of the competitors first translated the test from OWL to RDFS before running their algorithms. It is perfectly admissible that not all the benchmark campaign use the same languages.

Tasks involving multi-alignment are very specific. Indeed, usually alignment is triggered by editors that what to expand an ontology or web services to compose. This involves the alignment of two ontologies. Bringing other ontologies in the process does not help solving the problem. Multi-alignment is rather reserved to ontology normalisation or mining. For the moment it seems preferable to consider only two ontologies to align. This should hold until competitors complain that multi-alignment would be worthwhile.

## 2.2 Input alignment

The input alignment ($A$) can have the following characteristics:

**Complete/update:** Is the alignment process required to complete an existing alignment? (i.e., is $A$ non empty).

**Multiplicity** : How many entities of one ontology can correspond to one entity of the others? (see "Output alignment").

For the first kind of benchmark it seems reasonable that no input alignment will be provided. Of course, the competitors are free to deign their method around the composition of various methods which provide intermediate alignments.

## 2.3 Parameters

Parameters ($p$, $r$) of the alignment process were identified as:

**Oracles/resources** Are oracle authorized? If so, which ones (the answer can be any)? Is human input authorized?

**Training** Can training be performed on a sample?

**Proper parameters** Are some parameter necessary? And what are they? This point is quite important when a method is very sensitive the variation of parameters. A good tuning of these must be available.

Many systems take advantage of some external resources such as WordNet, sets of morphological rules or a previous alignment of general purpose catalogues (Yahoo and Google for instance). It is perfectly possible to use these resources as long as they have not be tuned to the task for the current benchmark (for instance, using a sub-lexicon which is dedicated to the domain considered by the tests). Of course, it is perfectly acceptable that the algorithms prune or adapt these resources to the actual ontologies. This is considered as the normal process of the algorithm. However this processing time must be considered within the running time of the algorithm.

Some algorithms could take advantage of the web for selecting some resource that is adapted to the considered ontology. This is perfect behaviour. However, as long as this is not specifically required by some competitor and because this is quite difficult to control, we think that this should not be authorised in the first place.

In general, if human input is provided, the performance of systems can be expected to be better. However, in the current state, which is the absence of any consensus or valuable methods for handling and evaluating the contribution of this human input, we will not take this into account.

Training on some sample is very often used by methods for aligning ontologies and mapping schemas. However, this training sample is a particular alignment. The only situation in which this makes a lot of sense is when a user provides some example of aligned instances and the system can induce the alignment from this. This is thus quite related to user input. We consider that this is an interesting characteristics to be considered in a second step.

Of course, some parameters can be provided to the methods participating on the evaluation. However, these parameters must be the same for all tests. It can be the case that some methods are able to tune their parameters depending on the presented ontologies. In such a case, the tuning process is considered part of the method. However, this process must be computed from the ontology input only, not from externally provided expected results.

It seems necessary, in competence benchmark, to have participants providing the best parameter set they found for the benchmark. This set must be the same for all tests. In competitive tests, especially when the expected result is not known from the participants, they will not change their parameters.

## 2.4 Output alignment

We identify the following possible constraints on the output alignment ($A'$) of the algorithm:

**multiplicity** How many entities of one ontology can correspond to one entity of the others? Usual notations are 1:1, 1:m, n:1 or n:m. We prefer to note if the mapping is injective, surjective and total or partial on both side. We then end up with more alignment arities (noted with, 1 for injective and total, ? for injective, + for total and * for none and each sign concerning one mapping and its converse): ?:?, ?:1, 1:?, 1:1, ?:+, +:?, 1:+, +:1, +:+, ?:*, *:?, 1:*, *:1, +:*, *:+, *:*. These assertions could be provided as input (or constraint) for the alignment algorithm or be provided as a result by the same algorithm.

**justification** Is a justification of the results provided?

**relations** Should the relations involved in the correspondences be only equivalence relations or could they be more complex?

**strictness** Can the result be expressed with trust-degrees different than $\top$ and $\bot$ or should they be strictified before?

In real life, there is no reason why two independently developed ontologies should have a particular alignment multiplicity other than *:*. This should be the (non) constraint on the output alignment of the benchmark tests. However, if we say so and all our tests provides some particular type of alignment (for instance, ?:? in the EON ontology tests), it can be said that this introduces a biais. This biais can be suppressed by having each type of alignment equally represented. However, this is not easy to find and this is not realistic. What would be realistic would be to have a statistical evaluation of the proportion of each type of alignment. In the absence of such an evaluation, however, it remains reasonable to stick to the *:* rule. This could be revised later on.

Another worthwhile feature for users is the availability of meaningful explanations or justifications of the correspondences. However, very few algorithms are able to deliver them and there is no consensus either on the form in which they are expressed neither on the way to compare them. So, it is currently not possible to ask for explanations in the benchmark results.

As mentioned in Deliverable 2.2.1 and 2.2.3, all algorithms deliver pairs of entities (called correspondences). However, some of them associate a relation between the entities different from equivalence (e.g., specificity) and some of them associate a strength to the correspondence (which can be a probability measure). A problem is that not all algorithms deliver the same structure. Moreover, alignments must be used in tasks for which, most of the time it is necessary to know how to interpret a term of one ontology with regard to another ontology. For these reasons, and because each method can, at least, deliver equivalence statement with the maximum strength, it seems better to avoid using any kind of relation or measure (more exactly, to design the tests with alignment involving only equivalence relations and $\top$ confidence measure.

## 2.5 Alignment process

The alignment process ($f$) itself can be constrained by:

**resource constraints** Is there a maximal amount of time or space available for computing the alignment?

**Language restrictions** Is the mapping scope limited to some kind of entities (e.g., only T-box, only classes)?

**Property** Must some property be true of the alignment? For instance, one might want that the alignment (as defined in the previous chapter be a conseqeunce of the combination of the ontologies (i.e., $o, o' \models A'$) or that alignments preserve consequences (e.g., $\forall \phi, \phi' \in L, \phi \models \phi' \implies A'(\phi) \models A'(\phi')$) or that the initial alignment is preserved (i.e., $o, o', A' \models A$).

Resource constraints can be considered either as a constraint (the amount of resource is limited) or a result (the amount consumed is measured – see Chapter 3). It is a relatively important factor, at least for performance tests and must be measured. This can also be measured for competence tests (even if it is absolutely difficult to do because of the heterogeneity of the environments in which these algorithms can be run).

Constraints on the kind of language construct to be found in mappings can be designed. However, currently very few alignment algorithms can align complex expressions, most of them align the identified (named) entities and some of them are only restricted to concepts. With regard to its importance and its coverage by current alignment systems, it makes sense to ask for the alignment of named entities and consider complex expressions later.

The properties of the alignments provided by the alignment algorithms are not very often mentioned and they seems to be very heterogeneous depending of the implemented techniques. It seems thus difficult to ask for particular properties. As for the type of alignment, not asking for a property is a problem if the tests do not satisfy a variety of properties. Moreover, it is not obvious that in real life, there are any properties to be satisfied by alignments (because ontologies are made for different purposes). So, at this stage, we do not commit to a particular property.

## 2.6 Conclusion

We propose to focus first on the simplest kind of test:

- comparing *two* ontologies written in the *same language*: OWL-DL;
- without input alignment;
- with any kind of fixed parameters and any kind of fixed and general purpose resources;
- without any kind of user input nor training samples;
- provide a strict *:* equivalence alignment of named entities;
- and measure the amount of resources consumed.

Like TREC has evolved towards multi-track competitions considering different benchmark set-up, it seems reasonable that the decision proposed here will have to be reconsidered with the evolution of the field.

It will then be natural to have extensions around the following features (ordered by perceived importance):

- considering another language than OWL;
- considering any kind of external resources (use of the web as it is);
- considering non-strict alignments and alignments with various types of relations;
- considering aligning with complex kind of expressions.

or specific tracks around (ordered by perceived importance):

- alignment with training on some sample seems a very important task;
- alignment with human input;
- alignment under difficult resource constraints (and even anytime alignment);
- alignments satisfying some formal properties;
- considering the alignment completion task;
- depending on task, consider more specific types of alignments (e.g., 1:1).

# Chapter 3

# Evaluation measures

This chapter is concerned with the question of how to measure the evaluation results returned by the benchmarking. It considers a wide range of different possible measures for evaluating alignment algorithms and systems. They include both qualitative and quantitative measures. We divide them into compliance measures which evaluate the degree of conformance of the alignment methods to what is expected, performance measures which measure non functional but important features of the algorithms (such as speed), user-related measures focusing on user evaluation, overall aggregating measures, and measures to evaluate specific tasks or applications.

## 3.1 Compliance measures

Compliance measures evaluate the degree of compliance of a system with regard to some standard. They can be used for computing the quality of the output provided by a system compared to a reference output. Note that such a reference output is not always available, not always useful and not always consensual. However, for the purpose of benchmarking, we can assume that it is desirable to provide such a reference.

### 3.1.1 Precision, recall, and others

There are many ways to qualitatively evaluate returned results [Do *et al.*, 2002]. One possibility consists of proposing a reference alignment ($R$) that is the one that the participants must find (a *gold standard*). Their results ($A$) from the evaluation runs can then be compared to that reference alignment. In what follows, the alignments $A$ and $R$ are considered to be sets of pairs.

The most commonly used and understood measures are precision (true positive/retrieved) and recall (true positive/expected) which have been adopted for ontology alignment. They are commonplace measures in information retrieval.

**Definition 2 (Precision).** *Given a reference alignment $R$, the precision of some alignment $A$ is given by*

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

Please note, that precision can also be determined without explicitly having a complete reference alignment. Only the correct alignments among the retrieved alignments have to be determined ($R \cap A$), thus making this measure a valid possibility for ex-post evaluations.

**Definition 3 (Recall).** *Given a reference alignment $R$, the recall of some alignment $A$ is given by*

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

The fallout measures the percentage of retrieved pairs which are false positive.

**Definition 4 (Fallout).** *Given a reference alignment $R$, the fallout of some alignment $A$ is given by*

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

Precision and recall are the most widely and commonly used measures. But usually, when comparing systems one prefers to have only one measure. Unfortunately, systems are often not comparable based solely on precision and recall. The one which has higher recall has lower precision and vice versa. For this purpose, two measures are introduced which aggregate precision and recall.

The F-measure is used in order to aggregate the result of precision and recall.

**Definition 5 (F-measure).** *Given a reference alignment $R$ and a number $\alpha$ between 0 and 1, the F-measure of some alignment $A$ is given by*

$$M_\alpha(A,R) = \frac{P(A,R) \cdot R(A,R)}{(1-\alpha) \cdot P(A,R) + \alpha \cdot R(A,R)}.$$

If $\alpha = 1$, then the F-measure is equal to precision and if $\alpha = 0$, the F-measure is equal to recall. In between, the higher $\alpha$, the more importance is given to precision with regard to recall. Very often, the value $\alpha = 0.5$ is used, i.e. $M_{0.5}(A,R) = \frac{2 \times P(A,R) \times R(A,R)}{P(A,R)+R(A,R)}$, the harmonic mean of precision and recall.

The overall measure (defined in [Melnik *et al.*, 2002] as accuracy) is an attempt of measuring the effort required to fix the given alignment (the ratio of the number of errors on the size of the expected alignment). Overall is always lower than the F-measure.

**Definition 6 (Overall).** *Given a reference alignment $R$, the overall of some alignment $A$ is given by*

$$O(A,R) = R(A,R) \times \Big(2 - \frac{1}{P(A,R)}\Big).$$

*It can also be defined as:*

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

When comparing systems in which precision and recall can be continuously determined, it is more convenient to draw the precision/recall curve and compare these curves. This kind of measure is widespread in the results of the TREC competitions.

### 3.1.2 Weighted Hamming distance

The Hamming distance measures the similarity between two alignments by counting the joint correspondences with regard to the correspondence of both sets.

**Definition 7 (Hamming distance).** *Given a reference alignment $R$, the Hamming distance between $R$ and some alignment $A$ is given by*

$$H(A,R) = 1 - \frac{|A \cap R|}{|A \cup R|}.$$

The Weighted Hamming distance pays attention not only to the correspondences but to their strengths as well. It requires that the strengths (as defined in deliverable D2.2.1) be the same in both sets of correspondences.

**Definition 8 (Weighted Hamming distance).** *Given a reference alignment $R$, the weighted Hamming distance between $R$ and some alignment $A$ is given by*

$$W(A,R) = \sum_{c \in A \cup R} \frac{|strength_A(c) - strength_R(c)|}{|A \cup R|}$$

*in which $strength_X(c)$ is $0$ if $c \notin X$.*

However, since the semantics of strength is not well defined, it is hazardous to use them for comparing alignments. Moreover, it can be considered that some reference alignment is always achievable in each context. In such a case, it would be useful to compare an exact (hardened) version of each obtained alignment rather than a rough alignment unless the way it is used is known.

It could be more interesting to measure from how far the alignment missed the target. To that extent it would be necessary to measure a distance from an obtained alignment and a reference alignment. However, this distance seems currently tricky to define for several reasons:

- it shall highly depend on the task to be performed;
- it will introduce a biais in the evaluation of the algorithms in favour of those based on this distance. This is not acceptable unless it is certain that this distance is the best one.

## 3.2 Performance measures

Performance measures (or non-functional measures) measure the resource consumption for aligning two ontologies. They can be used when the algorithms are 100% compliant or balanced against compliance [Ehrig and Staab, 2004]. Unlike the compliance measures, performance measures depend on the benchmark processing environment and the underlying ontology management system. Thus it is rather difficult to obtain objective evaluations.

### 3.2.1 Speed

Speed is measured in amount of time taken by the algorithms for performing their alignment tasks. If user interaction is required, one has to ensure to effectively measure the processing time of the machine only.

### 3.2.2 Memory

The amount of memory used for performing the alignment task marks another performance measure. Due to the dependency with underlying systems, it could also make sense to measure only the extra memory required in addition to that of the ontology management system (but it still remain highly dependent).

### 3.2.3 Scalability

There are two possibilities for measuring scalability, at least in terms of speed and memory requirements. First, it can be assessed by theoretical study. And second, it can be assessed by benchmark campaigns with quantified increasingly complex tests. From the results, the relationship between the complexity of the test and the required amount of resources can be represented graphically and the mathematical relationship can be approximated.

## 3.3 User-related measures

So far the measures have been machine focused. In some cases algorithms or applications require some kind of user interaction. This can range from the user utilizing the alignment results to concrete user input during the alignment process. In this case, it is even more difficult to obtain some objective evaluation. This subsection proposes measures to get the user into the evaluation loop.

### 3.3.1 Level of user input effort

In case algorithms require user intervention, this intervention could be measured in terms of some elementary information the users provide to the system. When comparing systems which require different input or no input from the user, it will be necessary to consider a standard for elementary information to be measured. This is not an easy task.

### 3.3.2 General subjective satisfaction

From a use case point of view it makes sense to directly measure the user satisfaction. As this is a subjective measure it cannot be assessed easily. Extensive preparations have to be made to ensure a valid evaluation. Almost all of the objective measures mentioned so far have a subjective counterpart. Possible measurements would be:

- input effort,
- speed,
- resource consumption (memory),
- output exactness (related to precision),
- output completeness (related to recall),
- and understandability of results (oracle or explanations).

Due to its subjective nature numerical ranges as evaluation result are less appropriate than qualitative values such as very good, good, satisfactory, etc.

## 3.4 Aggregated measure

Different measures suit different evaluation goals. If we want to improve our system, it is best to have as many indicators as possible. But if we want to single out the best system, it is generally easier to evaluate with very few or only one indicator. To allow for this, the different individual measurements have to be aggregated. This can be achieved by giving every measurement a weight

(e.g., in form of a weighted linear aggregation function). Obviously the weights have to be chosen carefully, again dependent on the goal.

**Definition 9 (Aggregated measure).** *Given a set of evaluation measures $m_i \in M$ and their weighting $w_i \in W$, the aggregated measure $Aggr$ is given by*

$$Aggr(M, W) = \sum_{m_i \in M} w_i \cdot m_i.$$

## 3.5 Task specific evaluation

So far evaluation was considered in general. But the evaluation could also be considered in the context of a particular task.

As a matter of fact, there are tasks which require high recall (for instance aligning as a first step of an interactive merge process) and others which require high precision (e.g. automatic alignment for autonomously connecting two web services). Different *task profiles* could be established to explicitly compare alignment algorithms with respect for certain tasks. The following short list of possible scenarios gives hints on such scenarios (taken deliverable 2.2.3):

- Agent communication,
- Emergent semantics,
- Web service integration,
- Data integration,
- Information sharing and retrieval from heterogeneous sources,
- Schema alignment or merging in overlay networks.

In terms of measurements, it would be useful to set up experiments which do not stop at the delivery of alignments but carry on with the particular task. This is especially true when there is a clear measure of the success of the overall task. Even without this, it could be useful to share corresponding aggregate measures associated to these "task profile".

Nevertheless, it will be extremely difficult to determine the evaluation value of the alignment process independently. The effects of other components of the overall application have to carefully filtered out.

## 3.6 Conclusion

This chapter presented several approaches to measure evaluations ranging from quality to resource consumption, from machine-focused to user-focused, and from general to task-specific measures.

However, it seems that currently the most natural factors to measure quality are precision and recall because they can be interpreted easily.

The next kind of measure to consider in upcoming benchmarking efforts are resource consumption and task-specific evaluations. Despite the different kinds of problems for the evaluation, which have to be overcome first, these measures are important for reaching the next steps of ontology alignment algorithms and should therefore be considered in very near future.

# Chapter 4

# Rules of evaluation

We have already shown, in experiment run this year, that we can do some evaluation in which people can relatively easily jump in, even within a short span of time. The results given by the systems make sense and certainly made the tool designers think. We plan to merge the two events which occurred this year (see Appendix).

The evaluation process (the rules of the game) must be defined beforehand. We consider here some possible ways to carry out alignment evaluation and propose to consider more specifically some of them.

We first consider the principles that must guide our evaluation effort before examining the TREC example and providing some rules for evaluating alignment algorithms based on these principles and our experience.

## 4.1 Principles

We describe below a number of principles that must guide the evaluation process. These principles will justify the rules below.

### 4.1.1 Continuity

The benchmarking must not be a one-shot exercise but requires continuous effort to identify the progress made by the fields (and eventually stop benchmarking when no more progress is made). This is endorsed by the "continous improvement" aspect of benchmarking.

These requires that it is carried out by some independent and sustainable entity.

### 4.1.2 Quality and equity

In order to be worthwhile, the evaluation campaign and material must be of the best possible quality. This also means that the benchmarking material must not biaised towards some particular kind of algorithm but driven by the tasks to solve.

It must be recognised among the community that is supposed to use and take advantage of them. People coming from different views with different kind of tools do not naturally agree on what is a good test.

In order to overcome this problem, the benchmark test must not be produced by only one entity and must be agreed by the major players. Moreover, automated as much as possible test generation and evaluation does provide a better chance to equity.

### 4.1.3 Dissemination

In order to have the most important impact, the evaluation activity must be disseminated without excessive barrier.

To that extent the benchmark tests and results must be published and certainly made freely available. The evaluation campaigns must not be restricted to the European research area: they must be open to participants worldwide. It could be important that these evaluation are announced in and reach as many different community as possible, not only the Semantic web community.

### 4.1.4 Intelligibility

It is of higher importance that the benchmark results could be analysed by the stakeholders and understood by everyone.

For that purpose, it is important that not only the final results be published but also the alignments themselves. Moreover, very important are the papers produced by participants commenting on their results.

### 4.1.5 Cost

In order to attract as many participants as necessary, the cost of participating must be low. The cost of organising and preparing the test must also be as low as possible.

For that purpose, the processes of generating the tests and running them must be as automated as possible.

## 4.2 Example of evaluation: TREC

TREC[1] is the "Text REtrieval Conference" organised by the NIST in the USA. It has been run yearly since 1992. It is a very good model for evaluation in a focussed research field, especially because it has been very successful.

TREC's goals are:

- increase research in information retrieval based on large-scale collection;
- provide a forum for stakeholders;
- facilitate technology transfer;
- improve evaluation methodology;
- create a series of test collections on various aspects of IR.

It is now organised in several tracks (corresponding to one kind of evaluation) which is organized over several years (5 is now the standard) for being able to compare the results. Tracks organized so far have covered:

- static text retrieval;

[1] http://trec.nist.gov

- interactive retrieval;
- information retrieval in a narrow domain using ad hoc resources (genomics);
- media (other than text) retrieval;
- answer finding.

Each track typically has between 8 and 20 participants. While each track is precisely defined, TREC has now a track record on investigating the evaluation of many different features of the retrieval task.

## 4.3 Sample rules

Here are sample and simple rules proposed for creating and running the evaluation of alignment algorithms. They are drawn from the principles above and our experience. They should be more precisely phrased out for each individual evaluation campaign.

### 4.3.1 Infrastructure

As presented before the evaluation must be run by some sustainable organisation. This organisation can be a legal entity or not but cannot be a research project. It can be associated with some agency (like NIST for TREC), some professional association (like ACM), some special purpose organisation (like SWSA for ISWC) or a totally informal but visible association (the Bourbaki group).

This organisation would have the main role of organising the evaluation campaigns, publicising them and ensuring the availability of their results.

Moreover, in order to achieve representativity and breadth, the evaluation must be organised by some committee. This committee is in charge of approving the rules of the campaigns, the benchmark tests and the results of the campaign.

The organisation must develop a permanent web site ensuring the availability of all benchmark tests, results and papers.

In order to be attractive for researchers and to ensure the archive service, it would be worthwhile to have a proceedings series at some publisher. Another idea that could be considered is to have an arrangement with some journal in order to fast track an extended version of the performance test winner's paper.

### 4.3.2 Campaigns

The idea of evaluation campaigns consists in holding a meeting at which (or previously to which), participants run their system on a well defined set of tests.

These campaigns could be run yearly and the meeting could be associated with various events (not always from the same community seems worth).

The initial architecture is to propose two compulsory tests improving on those designed for the EON Ontology Alignment Contests and I3CON events:

- a stable series of competence benchmark allowing to position the participants and assess the global evolution of the field. The results of these benchmarks should be available before the tests.

- a renewed "real-world" challenge playing the role of performance benchmark. The results of this challenge would only be uncovered at the meeting.

This architecture could evolve towards a track-structure in which the performance benchmark is replaced by several alternative tracks.

These tests can be produced by different teams. Their structure and processing terms must be clearly stated (in the way of the conclusion of Chapter 2).

The participants are required to provide their alignment results in a particular format. They are provided with software tools for helping them to produce their results and assess their performances before the meeting. Results to all tests are compulsory as well as a "constrained" paper describing the test processing. Participants are expected to produce a demonstration at the meeting.

The results of these tests would be evaluated by clearly announced measures (currently precision and recall are the measure of choice according to Chapter 3). Additional measures could be computed from the resulting alignment. The test could evolve towards additional measures.

### 4.3.3 Continuous process

With the availability of more automation, it will even be possible to provide continuous online submission of results (having thus a non-stop effort). In order to guarantee some evaluation of these results, they can be marked non validated when it is submitted and validated when, for instance, three members of the committee independently run the tests and received the same results. Of course, the burden would be on submitters to provide easy to set up and well-documented systems. This would help promote reproducibility.

Finally, in order to facilitate the participation to the contests, we must develop tools in which participants can plug and play their systems. In addition to the current evaluators and alignment loaders, we could provide some iterators on a set of tests for automating the process and we must automate more of the test generation process.

## 4.4 Conclusion

Based on the principles of continuity, quality, equity, dissemination, intelligibility and low entry cost and our experiment of organising and participating to evaluation efforts, we have proposed a first set of rules for organising a continuing benchmarking of ontology alignment. Of course, these rules must be refined and adapted but we think that they can really support alignment evaluation.

We will now consider the general guidelines for implementing some of these rules.

# Chapter 5

# Setting up the evaluation material

We consider here the task of setting up the evaluation material taken from our experience of organising two contests this year (see Appendix). This has no more interest than providing some hints about how this can be done.

We separate both sets of material from competence and performance benchmarks because they are done differently and because they can be carried out by different teams.

## 5.1 Competence benchmark

For the first competence experiment (the EON Ontology Alignment Contest), we designed a set of benchmark based on the following principle: start with one reference ontology and generate other ontologies to be aligned with it through systematic one by one discarding of language features found in the ontology[1]. This ensures a good coverage of all the situations in which an alignment algorithm can find itself and provides a good view of strengths and weaknesses of algorithms.

This set of systematic tests was complemented by two other series of tests: one considered simple tests such as comparing the ontology with an irrelevant one or with the ontology expressed in a weakened language. The last series of tests consisted in comparing it with as many relevant ontologies as we found in the web. This last series has basically the same requirement as those encountered in performance tests (hand-made alignment, etc.). We consider here only the systematic tests.

From the the 8 features considered in OWL ontologies as in our first experiment, it is possible to derive in such a way $2^8 = 256$ ontology pairs. As can be seen in Table 5.1, we used more than a single modality for altering a feature so that the number of tests would be far higher.

The number of tests is important. For that reason our first test campaign was not able to provide such a number of tests. It used only the base tests and we had the feeling that this somewhat biaised the results.

It is thus critical to be able to generate such tests semi-automatically (and to process them as well).

We provide below the description of all the tests as they where used in the first experiment, this should provide a raw blueprint on what is to be expected of such a competence benchmark tests. The subsection number is the number of the test in Table 5.1.

[1] In fact, as can be seen in Table 5.1, some tests considered several changes at once. This is because, class and property labels were initially considered as one category. Only tests discarding comments were not unit tests.

| Test | Class labels | Property labels | Comments | Subsumption hierarchy | Composition hierarchy | Restrictions | Properties | Instances |
|---|---|---|---|---|---|---|---|---|
| 101 | | | | | | | | |
| 201 | x | x | | | | | | |
| 201a | x | | | | | | | |
| 202 | x | x | x | | | | | |
| 202a | x | | x | | | | | |
| 203 | missp. | missp. | | | | | | |
| 204 | conv. | conv. | | | | | | |
| 205 | syn. | syn. | | | | | | |
| 206 | transl. | transl. | | | | | | |
| 221 | | | | x | | | | |
| 222 | | | | flat. | | | | |
| 223 | | | | exp. | | | | |
| 224 | | | | | | | | x |
| 225 | | | | | | x | | |
| 228 | | | | | | | x | |
| 230 | | | | | flat. | | | |
| | | | | | | | | |

Table 5.1: Table of tests and suppressed feature (test numbers are briefly described in the corresponding section). Abbreviations instead of cross corresponds to the method used for altering the ontology (*missp*elling, using different *conv*entions, *syn*onyms, *transl*ating, *flat*ening, and *exp*anding).

### 5.1.101 Identity

This simple test consists of aligning the reference ontology with itself.

### 5.1.201 No names

Each label or identifier is replaced by a random one.

### 5.1.202 No names, no comment

Each label or identifier is replaced by a random one. Comments (rdfs:comment and dc:description) have been suppressed as well.

#### 5.1.203 Misspelling of names

Each label or identifier is replaced by a missspelled one. Comments (rdfs:comment and dc:description) have been suppressed as well.

#### 5.1.204 Naming conventions

Different naming conventions (Uppercasing, underscore, dash, etc.) are used for labels. Comments have been suppressed.

#### 5.1.205 Synonyms

Labels are replaced by synonyms. Comments have been suppressed.

#### 5.1.206 Foreign names

The complete ontology is translated to another language than english (French in the current case, but other languages would be fine).

#### 5.1.221 No hierarchy

All subclass assertions to named classes are suppressed.

#### 5.1.222 Flattened hierarchy

A hierarchy still exists but has been strictly reduced.

#### 5.1.223 Expanded hierarchy

Numerous intermediate classes are introduced within the hierarchy.

#### 5.1.224 No instances

All individuals have been suppressed from the ontology.

#### 5.1.225 No restrictions

All local restrictions on properties have been suppressed from the ontology.

#### 5.1.226 No datatypes

In this test all datatypes are converted to xsd:string.

#### 5.1.227 Unit differences

(Measurable) values are expressed in different datatypes.

#### 5.1.228 No properties

Properties and relations between objects have been completely suppressed.

#### 5.1.229 Class vs instances

Some classes have become instances.

#### 5.1.230 Flattening entities

Some components of classes are expanded in the class structure (e.g., year, month, day attributes instead of date).

#### 5.1.231 Multiplying entities

Some classes are spread over several classes.

## 5.2 Performance benchmark

The first performance benchmarks organised this year aimed initially at aligning large-scale ontologies found on the web. However, it has been difficult to find such ontologies in the wild in a suitable form so that it can be used for a benchmark.

Moreover, these pairs of ontologies must be aligned for constituting a suitable contest. This is very difficult to find. Aligning by hand the ontology is necessary and this could bring all sorts of contestation (though it has not been the case in the final contest).

This problem is even more important if the ontology alignment must be a real blind competition in which the alignments are not known in advance because new alignments have to be produced for every new challenge. Moreover, the drastic change of benchmark prevent from any meaningful comparison of results of tests. This reinforce the importance of having constant competence benchmarks associated which could, at least, assess the non regression of the systems.

However, these performance benchmarks are very important because they use real world data in a competitive spirit. We provides here some tracks to set up such benchmarks and tasks.

### 5.2.1 Proposed sets

In order to help setting up such a benchmark in the following years we list some offers we have had from various teams with alignment of real world ontology pairs.

#### NIH

Olivier Bodenreider from the (US) National Institute of Health in Bedestha offered two major medical ontology/thesaurus that have been semi-automatically aligned by the National Library of Medicine [Zhang *et al.*, 2004]. The two ontologies are the Foundational model of anatomy FMA[2] (december 2004) and Galen[3] (reference model v6). FMA contains nearly 71000 concepts and 100 relationships and Galen 25000 concepts and 600 relationships. These ontologies do not

[2] http://sig.biostr.washington.edu/projects/fm/

[3] http://opengalen.org/

contain instances. Both ontologies do not have exactly the same coverage. FMA is expressed in a frame-like language and Galen in a description logic language[4]. Heiner Stuckenschmidt has also suggested to use the Tambis ontology[5] (but this one might be better to be compared with the GeneOntology[6]).

The alignment contains 3047 pairs of similar concepts, 340 pairs of candidate similar concepts. This alignment has also been considered for testing an alignment algorithm [Zhang *et al.*, 2005].

**Benefits** This is a large scale experiment with matters differing from evaluation with a very serious work done.

**Drawbacks** Ontologies certainly not in OWL, with a set of features more reduced.

### NII

Ryutaro Ichise from the (Japanese) National Institute of Informatics in Tokyo offered his alignments of Yahoo internet directory[7] (English, 2001 and 2004) and DMoz directory[8] (English version, 2001 and 2004) established from the common sites they index. These directories are not particularly ontologies. The 2004 repositories contains the documents categorised by the directories.

The alignments have been provided by Ryutaro in his papers [Ichise *et al.*, 2003; 2004], but the availability of indexed documents allows to build some objective measure of similarity (however, the intersection between the set of documents is relatively small so this is a problem for the acuracy of such a measure).

**Benefits** This is another large scale dataset.

**Drawbacks** These are not ontologies and are not carefully crafted. Yahoo directory is subject to copyright.

### IBM

Ramanathan Guha offered to help us with major ontologies to align. These are data sets from IBM and Microsoft BizTalk schemas.

**Benefits** Supposed to be large scale.

**Drawbacks** Not sure that it is not database schemas.

### VUA

Heiner Stuckenschmidt from Vrije Universiteit Amsterdam offered to provide a set of 47 aligned ontologies described by students about the same domain: academia [9]. He also has another set about TV schedules.

**Benefits** 47 ontologies are available so potentially $47^2 = 2209$ alignments!

[4] Our current understanding is that Galen is currently being converted to OWL and that the translation of FMA into OWL could be eased by its development under Protégé.

[5] http://www.ontologos.org/OML/..%5COntology%5CTAMBIS.htm

[6] http://www.geneontology.org/

[7] http://dir.yahoo.com/

[8] http://dmoz.com/

[9] http://wbkr.cs.vu.nl/

**Drawbacks** This is not a "real-world" test since the ontologies have been designed as an exercise by students. They are of a moderate size. These ontologies are in RDFS. The alignments are not available so far.

#### Politecnico de Milano

Laura Farinetti from the politecnico di Torino has several multilingual ontologies and would be willing to provide some for such an experiment.

**Benefits** Multilinguality for free. Already aligned by consensus committee.

**Drawbacks** Not sure that it is real ontologies. Are they in OWL?

#### Other possibilities

There is, of course, many different ontologies concerning our favorite experimental topic: the academy field that are around and that could be used.

As far as Knowledge web is concerned, having several ontologies used by semantic web services could be worthwhile.

The DAML ontology repository[10] does not seems to be used enough for being considered "real-world". Moreover, it does not really provide pairs of ontologies. The Illimois semantic integration archive [11] contains in fact data sets (and what is called an ontology is rather a directory or a hierarchy of instances). The Semantic integration resource[12] contains a few and limited aligned ontologies (in RDFS mainly).

### 5.2.2 Possible tasks

We mention here several specific tasks that could be considered in order to specialise the scope of benchmarks:

- aligning two ontologies;
- completing one ontology (with classes coming from another one);
- completing one ontology under a particular imposed subclass;
- finding what is necessary to translate a message or connect two web services.

## 5.3 Conclusions

We have a systematic way to generate some competence (or functionnal) benchmark tests. Automating their generation could help achieving higher coverage and more acuracy (see Chapter 6). This would also help concentrating on the performance benchmarks.

Performance benchmarks as described here are more difficult to develop because they are not systematic (and they thus require more manpower). We provided several opportunities that we could consider in order to build such benchmarks.

[10] http://www.daml.org/ontologies/

[11] http://anhai.cs.uiuc.edu/archive/

[12] http://www.atl.external.lmco.com/projects/ontology/

# Chapter 6

# Automation

Both evaluations we carried out shown that the job of participants and of running the evaluation were greatly facailitated by providing tools for the evaluation. The tools also have the good features of providing the results to the participants without ambiguity.

We present below both what is already available and how it is desirable to develop tools for evaluating ontology alignment algorithms.

## 6.1 Test generation framework

We did not use so far any test generation system. However, our competence benchmark would highly benefit from such systematic test generation facility. It is thus necessary to have some tools which, from one ontology, is able to discard any of the features and to generate both the obtained ontology and the corresponding alignment.

This generation facility could be relatively easy to provide for simple changes such as discarding entities or replacing labels by random strings. It is a bit more complicated when it must:

- replace by missspellings which would require a missspelling generator;
- translate terms which would require an automatic translation tool (some could be used for that);
- flatten subsumption and composition hierarchies which is however feasible;
- expand subsumption and composition hierarchies in a meaningful way which is far more difficult.

Such a generation tool would take some ontology as input and systematically generate directories corresponding to the combination of all the features considered by the competence benchmark and containing the altered ontology plus the corresponding reference alignment.

It could be useful to implement such a tool with interactive manipulations.

## 6.2 Alignment framework

The I3CON Experiement Set Platform is a workbench under which the participants who wanted it could adapt their tools and plug them in for generating the results. It also provided formats in n3 notation for alignments and measures.

The EON Ontology Alignment Contest made use of the Alignment API[3] for representing the resulting alignments. This API provide many different services (see [Euzenat, 2004]).

For using the Alignment framework, evaluation participants have to implement the Alignment API. The Alignment API enables the integration of the algorithms based on a minimal interface. Adding new alignment algorithms amounts to create a new `AlignmentProcess` class implementing the interface. Generally, this class can extend the proposed `BasicAlignment` class. The `BasicAlignment` class defines the storage structures for ontologies and alignment specification as well as the methods for dealing with alignment display. All methods can be refined (no one is final). The only method it does not implement is the one that implement the alignment algorithm: `align`. This method is invoked from the `Alignment` object which is already connected with the ontologies. I takes a `Parameters` structure enabling to communicate the parameters to the algorithms and must fill the `Alignment` object with the correspondence `Cells` that have been found by the algorithm.

Once this class (which can be thought of as a wrapper around the alignment algorithm) is implemented, it is used by creating an alignment object, providing the two ontologies, calling the `align` method which takes parameters and initial alignment as arguments. The alignment object then bears the result of the alignment procedure. It is thus possible to invoke it on a particular set of tests with particular parameters and to output the results on a variety of formats.

This will be exploited by launching the `GroupAlign` facility of the Alignment API package to align all pairs of ontologies in a list of subdirectories and generate the result in the required format in this directory.

## 6.3 Evaluation framework

The evaluation framework must enable the comparison of an alignment with another one and to generate a resulting evaluation. One of the available methods of the Alignment API (`PRecEvaluator`) directly provides precision, recall and F-measure in an extension of the format developed by Lockheed Martin.

Since the contest, the tools around the API have been improved. The first improvement consists of comparing the results of different algorithms simultaneously and generating a table. Other developments will consist in providing the opportunity to directly launch an algorithm to a full test bench (and even to optimise some parameter). We will try to merge both tools.

The evaluation framework is already implemented. It consists in gathering all the results in the same directory architecture and compare all of them to the reference alignment. This is implemented in the `GroupEval` class and has been used for the EON Ontology alignment contests (see Figure 6.1).

The Alignment API package provides a small utility (`GroupEval`) which allows to implement batch evaluation. It starts with a directory containing a set of subdirectories. Each subdirectory contains a reference alignment (usually called `refalign.rdf`) and a set of alignments (named `name1.rdf`...`namen.rdf`). These alignments can be provided directly by the `GroupAlign` facility.

Invoking `GroupEval` with the set of files to consider (-i argument) and the set of evaluation results to provide (-f argument with profm, for precision, recall, overall, fallout, f-measure as possible measures)

```
$ java -cp /Volumes/Phata/JAVA/ontoalign/lib/procalign.jar
    fr.inrialpes.exmo.align.util.GroupEval -f "pr" -c
```

```
-l "std,nea,ssda5,edna5,sdna5,karlsruhe,karlsruhe2,umontreal,fujitsu,stanford"
```

returns an HTML file (which could also be other format) such as the one for Figure 6.1.

| algo | karlsruhe2 | | umontreal | | fujitsu | | stanford | |
|---|---|---|---|---|---|---|---|---|
| test | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. |
| 101 | n/a | n/a | 0.59 | 0.97 | 0.99 | 1.00 | 0.99 | 1.00 |
| 102 | NaN | NaN | 0.00 | NaN | NaN | NaN | NaN | NaN |
| 103 | n/a | n/a | 0.55 | 0.90 | 0.99 | 1.00 | 0.99 | 1.00 |
| 104 | n/a | n/a | 0.56 | 0.91 | 0.99 | 1.00 | 0.99 | 1.00 |
| 201 | 0.43 | 0.51 | 0.44 | 0.71 | 0.98 | 0.92 | 1.00 | 0.11 |
| 202 | n/a | n/a | 0.38 | 0.63 | 0.95 | 0.42 | 1.00 | 0.11 |
| 204 | 0.62 | 1.00 | 0.55 | 0.90 | 0.95 | 0.91 | 0.99 | 1.00 |
| 205 | 0.47 | 0.60 | 0.49 | 0.80 | 0.79 | 0.63 | 0.95 | 0.43 |
| 221 | n/a | n/a | 0.61 | 1.00 | 0.98 | 0.88 | 0.99 | 1.00 |
| 222 | n/a | n/a | 0.55 | 0.90 | 0.99 | 0.92 | 0.98 | 0.95 |
| 223 | 0.59 | 0.96 | 0.59 | 0.97 | 0.95 | 0.87 | 0.95 | 0.96 |
| 224 | 0.97 | 0.97 | 0.97 | 1.00 | 0.99 | 1.00 | 0.99 | 1.00 |
| 225 | n/a | n/a | 0.59 | 0.97 | 0.99 | 1.00 | 0.99 | 1.00 |
| 228 | n/a | n/a | 0.38 | 1.00 | 0.91 | 0.97 | 1.00 | 1.00 |
| 230 | 0.60 | 0.95 | 0.46 | 0.92 | 0.97 | 0.95 | 0.99 | 0.93 |
| 301 | 0.85 | 0.36 | 0.49 | 0.61 | 0.89 | 0.66 | 0.93 | 0.44 |
| 302 | 1.00 | 0.23 | 0.23 | 0.50 | 0.39 | 0.60 | 0.94 | 0.65 |
| 303 | 0.85 | 0.73 | 0.31 | 0.50 | 0.51 | 0.50 | 0.85 | 0.81 |
| 304 | 0.91 | 0.92 | 0.44 | 0.62 | 0.85 | 0.92 | 0.97 | 0.97 |

Figure 6.1: The various alignments output in HTML

## 6.4 Conclusion

Providing formats have the advantage of being able to compute new measures and the consensus is latter made on a new evaluation method. The set of tools that we have presented help automating the generation of tests and evaluation of results. As can be seen from Figure 6.2, teh three presented functions should be able to automate generation, processing and evaluation of the alignment algorithm $f$ on the basis of ontology $O$. This has the advantage of decreasing the rate of

errors and with it the risk of contestation. This also lower the costs of generating a new set of tests or evaluating again the set of results. This helps managing the evaluations. But these tools also reduces the amount of work necessary from the participants to run the tests. They can thus concentrate on performing at best. Moreover, it enables the participants to run the evaluation of their results easily which helps them to report problems early and to improve their algorithms against the actual benchmark results.

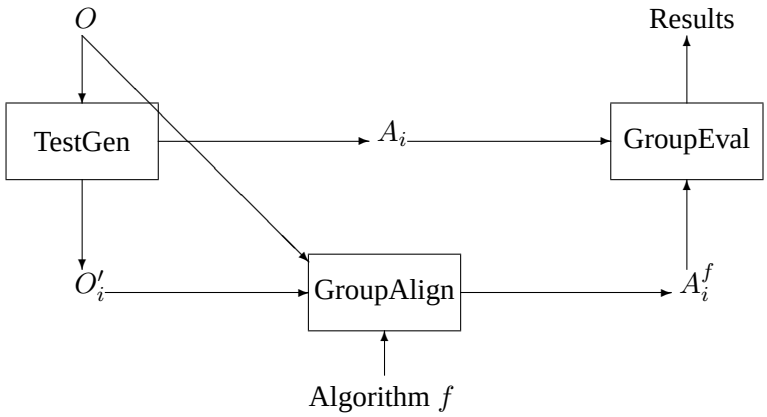


Figure 6.2: Process flow provided by the tool suite.

# Conclusion

We started from the goals of the evaluation (helping improving the state of the art in ontology alignment technology) and the definition of the alignment task. From this, and our experience of running and participating in two evaluation campaigns during the first year of Knowledge web, we have designed what we think should be a practical specification of such benchmarks.

It is based on a recurring yearly event combining the processing of a set of competence benchmark tests helping characterising the behaviour of each algorithm and a performance benchmark aiming at comparing algorithms performances on real world ontologies.

Our task in the coming months will consist in instantiating this framework and organising the second benchmarking campaign.

These benchmarks could, of course, also be useful in selecting the most appropriate tool for a particular task or application.

# Appendix: Evaluation experience

We present two different experiments that recently occured:

- The Information Interpretation and Integration Conference (I3CON), to be held at the NIST Performance Metrics for Intelligent Systems (PerMIS) Workshop, will be an ontology alignment demonstration competition on the model of the NIST Text Retrieval Conference. This contest focuses on "real-life" test cases and compare algorithm global performance. This effort has mainly been developed by Todd Hughes and Benjamin Ashpole of Lockheed Martin.
- The Ontology Alignment Contest at the 3rd Evaluation of Ontology-based Tools (EON) Workshop, to be held the International Semantic Web Conference (ISWC), will target the characterization of alignment methods with regard to particular ontology features. This initiative aims at defining a proper set of benchmark tests for assessing feature-related behavior. Because of its emphasis on evaluating the performances of tools instead of the competition between them, the term contest was not the best one. This contests has mostly been designed by Jérôme Euzenat at INRIA.

There was two different initiatives because the idea of evaluating alignment methods had been out since a long time [Noy and Musen, 2002; Do *et al.*, 2002] and there had been two occasions at once.

## First competitive evaluation (I3CON)

The I3CON[1] has been designed for providing some evaluation of ontology alignment algorithms.

The evaluation methodology consisted in publishing a set of ontologies to be compared with another ontology. As in the other test, the participants were asked to run one tool in one configuration on all the tests and to provide the results in a particular format. This format being similar but different from the previous one.

Contrary to the EON Ontology Alignment Contest, no reference alignment was provided, so the participant could not tune their system to find the best results for these tests.

A training set of two ontology pairs with their hand-made reference alignments was provided to the potential participant before the actual test cases so that they could adapt their systems for the contest.

The evaluation measure used were as usual: precision, recall and f-measure with regard to one secret reference alignment. No performance time measures were required.

A set of tool for running the tests was provided.

[1] `http://www.atl.external.lmco.com/projects/ontology/i3con.html`

### Test set

The set of tests was made of 8 ontology pairs. The ontologies concerned various domains (animals, Russia, soccer, basket ball, hotels, networks). The initial idea of the contest was to find ontology pairs on the web. However, this was not easy, so the organisers ended up by taking ontologies on the web and altering them. Various techniques have been used for the alteration (from random to adapting other ontologies concerning a related topic or using language translation).

The ontologies were provided in RDF/XML and n3, but their ontology language could be RDFS, DAML+OIL or OWL.

All the ontologies and reference alignments were produced by hand by consensus of an external team of students.

### Lesson learned from the first competition

The first good news of the I3CON experiment is that such a competition is possible.

For both evaluations, we expected five participants. There where five teams entering the I3CON initiative (ATL/Lockheed Martin, AT&T, INRIA, Karlsruhe and Teknowledge). This result is not too bad for a first run of experiment.

The results do not show a clear winner, some tests are more difficult to some algorithms than others. It is quite difficult to learn from the results of this experiments, and this reinforce the benefit of having papers from the participants.

The performance of the experiment itself showed that it was quite difficult to find pair of ontologies suitable for such a benchmark: either they are not comensurate, or they are expressed in different languages or the alignment does not exist. The ontology pairs have thus been, for most of them created for the purposes of the evaluation on a realistic basis.

Contrary to the EON Ontology Alignment Contest, participants did not use the tools provided to them. This could come from a lack of documentation, and should be investigated for further efforts.

## First competence benchmarks (EON Ontology Alignment Contest)

The EON "Ontology alignment contest"[2] has been designed for providing some evaluation of ontology alignment algorithms.

The evaluation methodology consisted in publishing a set of ontologies to be compared with another ontology. The participants were asked to run one tool in one configuration on all the tests and to provide the results in a particular format. In this format[3], an alignment is a set of pairs of entities from the ontologies, a relation supposed to hold between these entities and a confidence measure in the aligned pair. The tools could use any kind of available resources, but human intervention.

Along with the ontologies, a reference alignment was provided (in the same format). This alignment is the target alignment that the tools are expected to find. The reference alignment has all its confidence measures to the value 1 and most of the relations were equivalence (with very few subsumption relations). Because of the way the tests have been designed (see below), these alignments should not be contested. The participant were allowed to compare their results to the

[2] `http://co4.inrialpes.fr/align/Contest`
[3] `http://www.inrialpes.fr/exmo/software/ontoalign/`

output of their systems and the reference alignment and to chose the best tuning of their tools (overall).

The full test bench was proposed for examination to potential participants for 15 days prior to the final version. This allowed participants to provide some comments that could be corrected beforehand. Unfortunately, the real comments came later.

The results of the tests were expected to be given in terms of precision and recall of correspondences found in the produced alignment compared to the reference alignment. No performance time measures were required. The participants were also asked to provide a paper, in a predefined format, describing their tools, their results and comments on the tests.

Tools were provided for manipulating the alignments and evaluate their precision, recall and other measures[3].

### Test set

The set of tests consisted in one medium ontology (33 named classes, 39 object properties, 20 data properties, 56 named individuals and 20 anonymous individuals) to be compared to other ontologies. All ontologies were provided in OWL under its RDF/XML format.

This initial ontology was about a very narrow domain (bibliographical references). It was designed by hand from two previous efforts. It took advantage of other resources whenever they were available. To that extent the reference ontology refers to the FOAF (Friend-of-a-friend) ontology and the iCalendar ontology.

There were three series of tests:

- simple tests 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 that were obtained by discarding some features of the initial ontology leaving the remainder untouched. The considered features were (names, comments, hierarchy, instances, relations, restrictions, etc.). This approach aimed at recognising what tools really need. Our initial goal was to propose not just one feature discard but all the combinations of such. Unfortunately, we were unable to provide them before the launch of the contest.
- four real-life ontologies of bibliographic references that were found on the web and left untouched.

All the ontologies and reference alignments were produced by hand in a very short time. This caused a number of problems in the initial test base that were corrected later.

### Results

The results of the EON contest [Sure *et al.*, 2004] were globally higher than these of the I3CON certainly because of the way benchmark were made (all coming from the same source) with very identified and localised distortion.

As a first note, we expected five participants but finally only four entered (Stanford/SMI, Fujitsu, INRIA & UoMontréal and Karlsruhe). This is few, especially with regard to all the alignments algorithms out there. We hope that these four participants are the pionneer who will induce the others to put their work under comparison.

| algo | karlsruhe2 | | umontreal | | fujitsu | | stanford | |
|---|---|---|---|---|---|---|---|---|
| test | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. |
| 101 | n/a | n/a | 0.59 | 0.97 | 0.99 | 1.00 | 0.99 | 1.00 |
| 102 | NaN | NaN | 0.00 | NaN | NaN | NaN | NaN | NaN |
| 103 | n/a | n/a | 0.55 | 0.90 | 0.99 | 1.00 | 0.99 | 1.00 |
| 104 | n/a | n/a | 0.56 | 0.91 | 0.99 | 1.00 | 0.99 | 1.00 |
| 201 | 0.43 | 0.51 | 0.44 | 0.71 | 0.98 | 0.92 | 1.00 | 0.11 |
| 202 | n/a | n/a | 0.38 | 0.63 | 0.95 | 0.42 | 1.00 | 0.11 |
| 204 | 0.62 | 1.00 | 0.55 | 0.90 | 0.95 | 0.91 | 0.99 | 1.00 |
| 205 | 0.47 | 0.60 | 0.49 | 0.80 | 0.79 | 0.63 | 0.95 | 0.43 |
| 221 | n/a | n/a | 0.61 | 1.00 | 0.98 | 0.88 | 0.99 | 1.00 |
| 222 | n/a | n/a | 0.55 | 0.90 | 0.99 | 0.92 | 0.98 | 0.95 |
| 223 | 0.59 | 0.96 | 0.59 | 0.97 | 0.95 | 0.87 | 0.95 | 0.96 |
| 224 | 0.97 | 0.97 | 0.97 | 1.00 | 0.99 | 1.00 | 0.99 | 1.00 |
| 225 | n/a | n/a | 0.59 | 0.97 | 0.99 | 1.00 | 0.99 | 1.00 |
| 228 | n/a | n/a | 0.38 | 1.00 | 0.91 | 0.97 | 1.00 | 1.00 |
| 230 | 0.60 | 0.95 | 0.46 | 0.92 | 0.97 | 0.95 | 0.99 | 0.93 |
| 301 | 0.85 | 0.36 | 0.49 | 0.61 | 0.89 | 0.66 | 0.93 | 0.44 |
| 302 | 1.00 | 0.23 | 0.23 | 0.50 | 0.39 | 0.60 | 0.94 | 0.65 |
| 303 | 0.85 | 0.73 | 0.31 | 0.50 | 0.51 | 0.50 | 0.85 | 0.81 |
| 304 | 0.91 | 0.92 | 0.44 | 0.62 | 0.85 | 0.92 | 0.97 | 0.97 |

Table 6.1: Precision and recall results for each test

Below is the table of precision and recall results computed on the output provided by the participants with the help of the alignment API implementation.

Here are some consideration of the results obtained by the various participants. These are not statistically backed up and only corresponds to a rough analysis. More explanations are found in the papers presented by the participants.

**There were two groups of competitors...**

In this test, there are clear winners it seems that the results provided by Stanford and Fujitsu/Tokyo outperform those provided by Karlsruhe and Montréal/INRIA.

In fact, it can be considered that these constitute two groups of programs. The Stanford+Fujitsu programs are very different but strongly based on the labels attached to entities. For that reason they performed especially well when labels were preserved (i.e., most of the time). The Karlsruhe+INRIA systems tend to rely on many different features and thus to balance the influence of individual features, so they tend to reduce the fact that labels were preserved[4].

This intuition should be further considered in the light of more systematic tests which were planned but never made.

**... and indeed three groups of tests**

The results[5] of the I3CON inititative were relatively homogeneous in the sense that no algorithm was clearly outperforming the others in all tasks and no task what more difficult than others.

Without going through a throughout statistical analysis of the results, it seems that the separation between three sets of test that we presented (indicated by the first digit of their numbers) is significant for the participants as well.

- The first four tests were relatively easily handled by all participants. All programs showed there a better recall than precision (but not very significant).
- The systematic tests show more difficulty for the programs in general and for those of the second group in particular.
- The real life test were even more difficicult for both groups of participants.

## Lesson learned from the first benchmarking campain

The first good thing that we learnt is that it is indeed possible to run such a test.

Another lesson that we have learnt is that OWL is not that homogeneous when tools have to manipulate it. Parsers and API for OWL (e.g., Jena and OWL-API) are not really aligned in their way to handle OWL ontologies. This can be related to very small matters which can indeed render difficult entering the challenge. It is our expectation that these products will improve in the coming year. For the moment we modified the files in order to avoid these problems.

It is very difficult indeed to synthesize these results. This is true because there were different tests and not all go in the same direction. So aggregating the results can be done in many ways (e.g., averaging, global P/R, counting dominance). Moreover, these results are based on two measures, precision and recall, which are very easily understood but dual in the sense that increasing one often decreases the other. This means that one algorithms can have sometimes the same results

[4] It seems also that these two programs produced some artefact that should be easily eliminated

[5] `http://www.atl.external.lmco.com/projects/ontology/papers/I3CON-Results.pdf`

| algo | karlsruhe2 | | umontreal | | fujitsu | | stanford | |
|---|---|---|---|---|---|---|---|---|
| test | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. |
| 1xx | . | . | 0,57 | 0,93 | 0,99 | 1,00 | 0,99 | 1,00 |
| 2xx | 0,61 | 0,83 | 0,55 | 0,89 | 0,95 | 0,86 | 0,98 | 0,77 |
| 3xx | 0,90 | 0,56 | 0,37 | 0,56 | 0,66 | 0,67 | 0,92 | 0,72 |
| total | 0,73 | 0,72 | 0,51 | 0,82 | 0,89 | 0,84 | 0,97 | 0,80 |

Table 6.2: Average value of precision recall per groups of tests and globally

as another but they are found non comparable in the table. As an indication, the average values are given in Figure 6.2.

It is noteworthy that for solving this problem, TREC uses curves on the precision$\times$recall space. But we do not now yet what is the protocol for obtaining such curves (may be by asking some ranking of the answers).

People appreciated to be given tools to manipulate the required formats. It is clear that in order to attract participants, the test process should be easy.

We also realised that the production of an incomplete test bench (not proposing all combinations of discarded features) had an influence on the result. As a matter of fact, algorithms working on one feature only were advantaged because in most of the tests this feature was preserved.

Another lesson we learned is that asking for a detailed paper was a very good idea. We have been pleased of how much insight can be found in the comments of the competitors.

# Bibliography

[Bouquet *et al.*, 2004] Paolo Bouquet, Jérôme Euzenat, Enrico Franconi, Luciano Serafini, Giorgos Stamou, and Sergio Tessaris. Specification of a common framework for characterizing alignment. deliverable D2.2.1, Knowledge web NoE, 2004.

[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.

[Ehrig and Staab, 2004] Marc Ehrig and Steffen Staab. QOM - quick ontology mapping. In *Proc. 3rd ISWC, Hiroshima (JP)*, November 2004. to appear.

[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, 2003] Jérôme Euzenat. Towards composing and benchmarking ontology alignments. In *Proc. ISWC-2003 workshop on semantic information integration, Sanibel Island (FL US)*, pages 165–166, 2003.

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

[Ichise *et al.*, 2003] Ryutaro Ichise, Hideaki Takeda, and Shinichi Honiden. Integrating multiple internet directories by instance-based learning. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 22–30. Morgan Kaufmann, 2003.

[Ichise *et al.*, 2004] Ryutaro Ichise, Masahiro Hamasaki, and Hideaki Takeda. Discovering relationships among catalogs. In Einoshin Suzuki and Setsuo Arikawa, editors, *Discovery Science*, volume 3245 of *Lecture Notes in Computer Science*, pages 371–379. Springer, 2004.

[Kalfoglou and Schorlemmer, 2003] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.

[Melnik *et al.*, 2002] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: a versatile graph matching algorithm. In *Proc. 18th International Conference on Data Engineering (ICDE), San Jose (CA US)*, 2002.

[Noy and Musen, 2002] Natasha Noy and Mark Musen. Evaluating ontology-mapping tools: requirements and experience. In *Proc. 1st workshop on Evaluation of Ontology Tools (EON2002), EKAW'02*, 2002.

[Rahm and Bernstein, 2001] Erhard Rahm and Philip Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[Stefani *et al.*, 2003] F. Stefani, D. Macii, A. Moschitta, and D. Petri. Fft benchmarking for digital signal processing technologies. In *17th IMEKO World Congress*, Dubrovnik, Croatia, 22-27 June 2003.

[Sure *et al.*, 2004] York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the 3rd Evaluation of Ontology-based tools (EON)*, 2004.

[Zhang *et al.*, 2004] Songmao Zhang, Peter Mork, and Olivier Bodenreider. Lessons learned from aligning two representations of anatomy. In *Proc. 13th KR*, pages 555–560, 2004.

[Zhang *et al.*, 2005] Songmao Zhang, Peter Mork, Olivier Bodenreider, and Philip Bernstein. Comparing two approaches for aligning representations of anatomy. *Artificial Intelligence in Medicine*, 2005. Submitted.

# 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 mentionned 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. |