Towards a methodology for evaluating alignment and matching algorithms
Version 1.0

Ontology Alignment Evaluation Initiative[1]

http://oaei.inrialpes.fr

May 2, 2005

[1]Coordinator: Jérôme Euzenat (INRIA Rhône-Alpes) with contributions of Marc Ehrig (Universität Karlruhe), and Raúl García Castro (UP Madrid)

Abstract

This document considers the potential strategies for experimentally evaluating ontology alignment algorithms. It first identifies various goals for such an evaluation, the most important objective being the improvement of existing methods. It considers the various parameters of the alignment task that must be controlled during the experiment and examine the measures that can be used for an evaluation. It then propose a framework for organising the evaluation based on some principles and efforts that have already been undergone in the specific field of ontology alignment.

Executive Summary

Heterogeneity problems on the semantic web can be solved, for some of them, by aligning or matching heterogeneous ontologies. Aligning ontologies consists of finding the corresponding entities in these ontologies. 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 report 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. It should be considered as a white paper describing what the Ontology Alignment Evaluation Initiative is supposed to be.

In this 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).

The description of the retained systematic competence benchmark tests as well as the effective rules of the evaluation campaigns will be the subject of other documents.

Contents

Chapter 1

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

When applications and agents use heterogeneous ontologies, it is necessary to reconciliate these ontologies before they interoperate. This can be achieved through the alignment or matching of the ontologies.

Aligning ontologies consists of finding the corresponding entities in these ontologies. There have been many different techniques proposed for implementing this process. 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.*, 2004a]. 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 for evaluating of these methods. Beside their apparent heterogeneity, it seems sensible to characterise an alignment as a set of pairs expressing the correspondences between two ontologies. Such a characterization should enable the comparison of the results provided by the algorithms.

However, very few experimental comparison of algorithms are available. It is thus one of the objectives of the Ontology Alignment Evaluation Initiative to run such an evaluation.

Since all benchmarking activity must be carried out with a systematic proce-

dure on clearly defined tasks, this is the purpose of this report to propose such a procedure. This introduction will define the general objective of evaluating the alignment algorithms, the overall methodology to be followed, and the kind of tests which can be performed.

1.1 Goal of evaluation

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 Evaluation methodology

Each evaluation must be carried out according to some methodology. [Castro *et al.*, 2004] presents a benchmarking methodology that is briefly summarized here.

1.2.1 Benchmarking features

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). 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: benchmarks usually measure the distance between a given result with some expected result (the distance can well be a yes or no answer);
– continuous improvement: benchmarks should allow to monitor the improvement (and non degradation) of a solution by running the benchmark tests again;
– systematic procedure: benchmark results must be non ambiguous and their procedure reproducible.

1.2.2 Benchmarking lifecycle

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 benchmarking process itself (see Figure 1.1). 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 clearly refining the goal and the subject of the evaluation, preparing the proposal for benchmarking, finding other organisations that want to participate in the benchmarking activity, 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.

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 report, mainly concerned with the design of the evaluation, i.e., the Plan and Experiment phases described above. Precising how to report and communicate on the results is considered in §4; while planning the corrective methods,

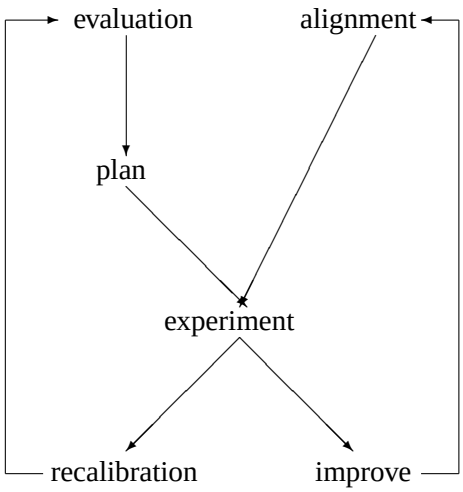

Figure 1.1: The evaluation process paralleled with the development process.

improving the actual systems and monitoring the results is a matter concerning algorithms developers and is not covered in this report. However, since, we already run two evaluation events in 2004, we have implemented these two steps already and this report can be seen as the end of the recalibration phase.

The remainder of this report consists in proposing mainly the main outline for the Plan phase.

1.3 Examples of evaluations

In order to illustration what can be done as evaluation, we briefly present a model evaluation initiative, TREC, and the two event that we organised in 2004.

1.3.1 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 goals are:

– increase research in information retrieval based on large-scale collection;

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

– 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;
– 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.

1.3.2 I3CON and EON

We have organised 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 [Sure *et al.*, 2004] and [Euzenat *et al.*, 2004b].

1.4 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 [Castro *et al.*, 2004], 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.

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.

1.4.1 Competence benchmark

Competence benchmarks aim at characterising the kind of task each method is good at or which kind of input it can handle well. 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.

1.4.2 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.).

1.5 Conclusion

The main goal of the Ontology Alignment Evaluation Initiative is the improvement of ontology alignment 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.

The conclusion of each chapter recalls the current options retained by the Ontology Alignment Evaluation Initiative.

This document is largely based on Knowledge web[2] deliverable 2.2.2 [Euzenat *et al.*, 2004b]. However, it will evolve independently in function of the Ontology Alignment Evaluation Initiative.

[2] http://knowledgeweb.semanticweb.org

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.

In [Bouquet *et al.*, 2004], we characterised 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.

[Bouquet *et al.*, 2004] 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.

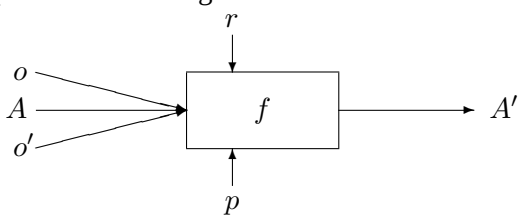

Figure 2.1: The alignment process.

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? (I.e., an alignment between more than two ontologies).

Currently, we consider 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. In Knowledge web 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 want 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 design their methods around the composition of various methods which provide intermediate alignments.

2.3 Parameters and resources

Parameters (p) and resources (r) of the alignment process are 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 been tuned to the purpose of 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 in 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. [Euzenat, 2003] prefers 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? (e.g., subsumption $\leq$, incompatibility $\perp$).

Strictness Can the result be expressed with trust-degrees different than $\top$ and $\perp$ 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 [Bouquet *et al.*, 2004] and [Euzenat *et al.*, 2004a], 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 benchmarking related 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 samples 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 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 Conformance measures

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*). The result from the evaluated alignment algorithm (A) can then be compared to that reference alignment. In what follows, the alignments A and R are considered to be sets of pairs.

A first simple distance between two sets is the Hamming distance measures the disimilarity between two alignments by counting the joint correspondences with regard to the correspondence of both sets.

Definition 2 (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 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 3 (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 4 (Recall). *Given a reference alignment R, the recall of some alignment A is given by*

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

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

Definition 5 (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 6 (F-measure). *Given a reference alignment R and a number α 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 α, 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 (also 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 7 (Overall). *Given a reference alignment R, the overall of some alignment A is given by*

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

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 Non equal correspondences

Currently, the proposed compliance measures are purely related to the identity of the correspondences (including strength and relation). It is possible to relax this constraint by just considering the couple of entities (disregarding the strength and relations).

This is not satisfactory because this does not account for the semantics of the relations and strengths. In order to relax this constraint, it is necessary to be able to measure some distance between strengths and relations.

The distance between the strength of two correspondences can be considered to be the absolute value between the two strength values. This can help comparing two set of correspondences on the basis of the strengths attributed to each correspondence.

Definition 8 (Strength-based distance). *Given a reference alignment R, the strength-based distance between R and some alignment A is given by*

$$SBD(A,R) = \sum_{c \in A \dot{\cap} R} |strength_A(c) - strength_R(c)|$$

in which $A \dot{\cap} R = \{\langle e, e', r, n_A, n_R\rangle; \langle e, e', r, n_A\rangle \in A \wedge \langle e, e', r, n_R\rangle \in R\}$ and such that $\forall e, e', r, \langle e, e', r, 0\rangle \in X$.

It is noteworthy that the strength-based distance can be used instead of the intersection in each of the definitions of § 3.1.1 (this has been done for Hamming distance in [Euzenat *et al.,* 2004b]).

Distances between relations have to be defined in some more qualitative way (one possibility could be the use of a graph distance on conceptual neighborhoods).

3.1.3 Measuring near misses

One difficulty with the previous measures is that they require that the exact same correspondence is in the alignment. 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 measures

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.

F-measure is already an aggregation of two measures (precision and recall). It can be generalized for any number of measures. This requires to attribute every measurement a weight (such that these weights sum to 1).

Definition 9 (M-measure). *Given a reference alignment R, a set of measures $(M_i)_{i\in I}$ provided with a set of weights $(w_i)_{i\in I}$ between 0 and 1 such that their sum is 1, the M-measure of some alignment A is given by*

$$M(A,R) = \frac{\prod_{i\in I} M_i(A,R)}{\sum_{i\in I} w_i \cdot M_i(A,R)}.$$

This also can be achieved by a weighted linear aggregation function. Obviously the weights have to be chosen carefully, again dependent on the goal.

Definition 10 (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 [Euzenat *et al.*, 2004a]):

- 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.

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

Organizational guidelines

Experiments run in 2004 have shown that we can do some evaluation in which people can relatively easily jump, 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 2004 events.

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 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 field (and eventually stop benchmarking when no more progress is made). This is endorsed by the "continous improvement" aspect of benchmarking.

These requires that benchmarking 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 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 Access 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 Proposed implementation

Here are sample and simple regulation 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 (see the rules for the EON Ontology alignment contest for instance).

4.2.1 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.

4.2.2 Campaigns

The idea of evaluation campaigns consists of 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.

4.2.3 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.

4.3 Conclusion

Based on the principles of continuity, quality, equity, dissemination, intelligibility and accessibility and our experience in 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.
The current structure for ensuring continuity is an informal organisation committee under the name of "Ontology Alignment Evaluation Initiative". It is currently not part of other effort but partly supported by the Knowledge web European network of excellence.

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 in 2004, 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. More precisely this will involve:

- creating a real world test case;
- completing the systematic competence benchmarks (through automating);
- setting the formal rules of the evaluation;
- launching the evaluation;
- revising this proposal.

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.

[Castro *et al.*, 2004] Raúl García Castro, Diana Maynard, Doug Foxvog, Holger Wache, and Rafael González-Cabero. Specification of a methodology, general criteria, and benchmark suites for benchmarking ontology tools. deliverable D2.1.4, 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.*, 2004a] Jérôme Euzenat, Thanh Le Bach, Jesús Barrasa, Paolo Bouquet, Jan De Bo, Rose Dieng-Kuntz, Marc Ehrig, Manfred Hauswirth, Mustafa Jarrar, Rubén Lara, Diana Maynard, Amedeo Napoli, Giorgos Stamou, Heiner Stuckenschmidt, Pavel Shvaiko, Sergio Tessaris, Sven Van Acker, and Ilya Zaihrayeu. State of the art on ontology alignment. deliverable D2.2.3, Knowledge web NoE, 2004.

[Euzenat *et al.*, 2004b] Jérôme Euzenat, Marc Ehrig, and Raúl García Castro. State of the art on ontology alignment. deliverable D2.2.2, 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.

[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.

[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.