OLA in the OAEI 2005 alignment contest

Jérôme Euzenat, Philippe Guégan, Petko Valtchev

OLA in the OAEI 2005 alignment contest

Jérôme Euzenat
INRIA Rhône-Alpes
Montbonnot, France
euzenat-at-inrialpes.fr

Philippe Guégan
DIRO, Université de Montréal
Montr eal (Qc), Canada
Guegan-at-iro.umontreal.ca

Petko Valtchev
DIRO, Université de Montréal
Montr eal (Qc), Canada
Petko.Valtchev-at-umontreal.ca

ABSTRACT

Among the variety of alignment approaches (e.g., using machine learning, subsumption computation, formal concept analysis, etc.) similarity-based ones rely on a quantitative assessment of pair-wise likeness between entities. Our own alignment tool, OLA, features a similarity model rooted in principles such as: completeness on the ontology language features, weighting of different feature contributions and mutual influence between related ontology entities. The resulting similarities are recursively defined hence their values are calculated by a step-wise, fixed-point-bound approximation process. For the OAEI 2005 contest, OLA was provided with an additional mechanism for weight determination that increases the autonomy of the system.

1. PRESENTATION OF THE SYSTEM

OLA (for *O*WL-*L*ite *A*lignment) is an open-source tool jointly developed by teams at University of Montréal and INRIA Rhône Alpes. It features similarity-based alignment and a set of auxiliary services supporting the manipulation of alignment results [5, 6].

1.1 General purpose statement

The primary goal behind the OLA tool design is to perform alignment of ontologies expressed in OWL, with a short-term emphasis on OWL-Lite and long-term one on OWL-DL. However, its GUI component, VisOn[1] allows for many other services involving alignments (in the sense of [4]) to be accessed.

1.1.1 Functional specifications

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

- parsing and visualization of OWL-Lite and OWL-DL ontologies,
- computation of similarities between entities from two ontologies,
- extraction of alignments from a pair of ontologies, provided with a set of similarity matrices, one per category of ontology entities (see below),
- manual construction of alignments by composing entity pairs from two ontologies,
- use of an existing (partial) alignment as a seed for automated alignment construction (alignment completion),
- alignment visualization,
- comparison of two alignments.

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

1.1.2 Principles of the alignment in OLA

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

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

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

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

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

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

1.1.3 Current limitations

- Although it would be of certain value for alignment, OLA currently offers no inference mechanisms that could help complete the entity descriptions. In particular, inheritance is not used to expand entities, mostly out of efficiency considerations.

- Although neighborhoods play crucial role in the similarity definition, two neighbor entities are not necessarily affecting each other's respective similarities to a pair of other entities. As only descriptive knowledge is taken into account, given two such entities, say e_1 and e_2, for e_2 to appear in a similarity expression for e_1, it should be considered as part of the description of the latter. For instance, a data type is not seen as being described by a property whose range the datatype represents. Consequently, datatypes are compared in an ontology-independent manner.

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

1.2 Specific techniques used

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

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

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

1.2.1 OL-Graph construction

OL-Graphs are graph structures that provide an easy-to-process inner representation of OWL ontologies. An OL-Graph is a labeled graph where vertices correspond to OWL entities and edges to inter-entity relationships. As described in [6], the set of different vertex categories is: class (C), object (O), relation (R), property (P), property instance (A), datatype (D), datavalue (V), property restriction labels (L). Furthermore, we distinguish between datatype relations (R_{dt}) and object relations (R_o), and between datatype properties (P_{dt}) and object ones (P_o).

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

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

The OL-Graph of an ontology is built after the ontology is parsed[4]. The process of OL-Graph construction is described in [8].

1.2.2 Similarity model

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

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

[4]So far, we use the OWL API [1].

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

Table 1: Similarity function decomposition (`card` = `cardinality` and `all` = `allValuesFrom`).

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

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

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

The function is normalized, i.e., the weights $\pi^X_{\mathcal{F}}$ sum to a unit, $\sum_{\mathcal{F} \in \mathcal{N}(X)} \pi^X_{\mathcal{F}} = 1$. for the computability The set functions $MSim_Y$ compare two sets of nodes of the same category (see [6] for details). Table 1 illustrates the set of similarities in our model.
Following the lessons learned with our participation in the EON 2004 alignment contest [**?**], we have adapted the above measure to fit cases where particular pair of entities is described only by a small subset of the entire set of category descriptors. Thus, a descriptive factor is ignored for similarity computation whenever neither of the compared entities possesses a neighbor with the underlying link label (e.g., no instances for a pair of compared classes). In this case, not only its weight is set to 0, but also the weights of the remaining "active" factors are increased correspondingly. To scale that principle up to the entire set of descriptive factors, the following simple mechanism has been realized in OLA: In order to keep both normalization and equity in similarity values, the weights of all non-null factors for a given entity pair are divided through their sum. Thus, for a category X, the similarity measure $Sim^+_X : X^2 \to [0,1]$ becomes:

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

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

OLA relies on various functions for identifiers comparison. Both string distances and lexical distances are used. Lexical distances rely on an exploration of WordNet 2.0 [7] with a quantitative assessment of the "relatedness" between two, possibly multi-word, terms. More specifically, the degree of relatedness between two WordNet entries is computed as the ratio between the depth, in graph-theoretic sense, of the most specific common hypernym and the average of both term depths. The computation of multi-word term similarity consists in first splitting the terms into a set of tokens each and then comparing all possible pairs of tokens from opposite sets using the above depth-based principle. The global term similarity is then computed as a similarity-based matching between both sets (see above).

As circular dependencies are impossible to avoid with the above definitions, the computing of the similarity values requires non-standard mechanisms. Following [2, 9], an equation system is composed out of the similarity definitions where variables correspond to similarities of node pairs while coefficients come from weights. The process of iterative, fixed-point-bound resolution of that system, as well as the related convergence and determinism issues are described in [6].

1.3 Implementation

OLA is implemented in JAVA. Its architecture follows the one of the Alignment API and the recent implementation that was described in [4]. OLA relies on the OWL API [1] for parsing OWL files. An entire subsystem is dedicated to the onstruction of OL-Graphs on top of the parsed ontologies. A set of further components that offer similarity computation services: substring distances, edit distances, Hamming distance, WordNet interface (via the JWNL library [3]), etc., that were originally designed for OLA are now part of the Alignment API. The VISON GUI component offers a uniform interface to all services provided by Alignment API and OLA. In particular, it visualizes both the input data, i.e., the OL-Graphs, and the final result, i.e., the alignment file, of the global process.

1.4 Adaptations made for the contest

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

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

1.4.1 Weight computing mechanism

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

1.4.2 Similarity measure for entity names

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

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

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

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

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

[6]subString distance provided by the Alignment API

1.4.3 Minor adaptations

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

2. RESULTS

The comments are grouped by test categories.

2.101 Tests 10X

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

2.102 Tests 2XX

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

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

The second groups is made of the tests 25X. Here OLA performances varied substantially: from extremely poor

(254) to satisfactory (252, 259).

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

2.103 Tests 30X

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

3. GENERAL COMMENTS

3.1 Comments on the results

The results show a substantial progress has been made since the EON 2004 alignment contest. With respect to the performances of OLA at that forum, we made a big leap amounting to about 25% in both mean precision and mean recall.

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

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

3.2 Discussions on the way to improve the proposed system

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

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

3.3 Comments on the experiment

Two months during summer period is definitely too short to run shuch an experiment.

4. CONCLUSION

In its latest version, OLA has proven a more robust tool for alignment than it was a year before. While the difficulties with real-world ontologies persist, the progress on noisy ones has been substantial.

The next key topic of the research around OLA will be the automation of the weight computation for a specific pair of ontologies.

5. RAW RESULTS

5.1 Link to the set of provided alignments

A .zip archive of all the contest results is available at the following URL:

`http://www.iro.umontreal.ca/~owlola/OAEI.html`

5.2 Link to the system and parameters file

A similar archive with the parameters and the .jar files used in the contest-related experiments is available at the following URL:

`http://www.iro.umontreal.ca/~owlola/OAEI.html`

5.3 Matrix of results

#	Name	Prec.	Rec.	Time
101	Reference alignment	1	1	
103	Language generalization	1	1	
104	Language restriction	1	1	
201	No names	0.71	0.62	
202	No names & no comments	0.66	0.56	
203	No comments	1	1	
204	Naming conventions	0.94	0.94	
205	Synonyms	0.43	0.42	
206	Translation	0.94	0.93	
207		0.95	0.94	
208		0.94	0.94	
209		0.43	0.42	
210		0.95	0.94	
221	No specialisation	1	1	
222	Flatenned hierarchy	1	1	
223	Expanded hierarchy	1	1	
224	No instance	1	1	
225	No restrictions	1	1	
228	No properties	1	1	
230	Flattened classes	0.95	0.97	
231		1	1	
232		1	1	
233		1	1	
236		1	1	
237		0.97	0.98	
238		0.99	0.99	
239		0.97	1	
240		0.97	1	
241		1	1	
246		0.97	1	
247		0.97	1	
248		0.59	0.46	
249		0.59	0.46	
250		0.3	0.24	
251		0.42	0.3	
252		0.59	0.52	
253		0.56	0.41	
254		0.04	0.03	
257		0.25	0.21	
258		0.49	0.35	
259		0.58	0.47	
260		0.26	0.17	
261		0.14	0.09	
262		0.2	0.06	
265		0.22	0.14	
266		0.14	0.09	
301	Real: BibTeX/MIT	0.42	0.38	
302	Real: BibTeX/UMBC	0.37	0.33	
303	Real: Karlsruhe	0.41	0.49	
304	Real: INRIA	0.74	0.66	

6. REFERENCES

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

[2] Gilles Bisson. Learning in FOL with similarity measure. In *Proc. 10th AAAI, San-Jose (CA US)*, pages 82–87, 1992.

[3] John Didion. The Java WordNet Library, http://jwordnet.sourceforge.net/, 2004.

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

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

[6] Jérôme Euzenat and Petko Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proc. 15th ECAI*, pages 333–337, Valencia (ES), 2004.

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

[8] Mohamed Touzani. Alignement d'ontologies dans OWL. Master's thesis, University of Montréal, in preparation.

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