OLA in the OAEI 2007 Evaluation Contest

Jean François Djoufak Kengue[1], Jérôme Euzenat[2], and Petko Valtchev[1]

[1] LATECE, Université du Québec À Montréal, Canada
[2] INRIA Rhône-Alpes and LIG, Montbonnot, France
djoufak.jean_francois@courrier.uqam.ca, jerome.euzenat@inrialpes.fr, valtchev.petko@uqam.ca

Abstract. Similarity has become a classical tool for ontology confrontation motivated by alignment, mapping or merging purposes. In the definition of an ontology-based measure one has the choice between covering a single facet (e.g., URIs, labels, instances of an entity, etc.), covering all of the facets or just a subset thereof. In our matching tool, OLA, we had opted for an integrated approach towards similarity, i.e., calculation of a unique score for all candidate pairs based on an aggregation of all facet-wise comparison results. Such a choice further requires effective means for the establishment of importance ratios for facets, or weights, as well as for extracting an alignment out of the ultimate similarity matrix. In previous editions of the competition OLA has relied on a graph representation of the ontologies to align, OL-graphs, that reflected faithfully the syntactic structure of the OWL descriptions. A pair of OL-graphs was exploited to form and solve a system of equations whose approximate solutions were taken as the similarity scores. OLA2 is a new version of OLA which comprises a less integrated yet more homogeneous graph representation that allows similarity to be expressed as graph matching and further computed through matrix multiplying. Although OLA2 lacks key optimization tools from the previous one, while a semantic grounding in the form of WORDNET engine is missing, its results in the competition, at least for the benchmark test suite, are perceivably better.

1 Introduction

Ontologies, i.e., explicit conceptualizations of a domain involving representations of domain concepts and relations, are now the standard way to approach data heterogeneity on the Web and insure application *interoperability*. However, the existence of independently built ontologies for the same domain is a source of heterogeneity on its own and therefore calls for the design of methods and tools restoring interoperability through ontology *matching*.

Similarity has become a classical tool for ontology matching. In the definition of an ontology-based similarity measure one has the choice between covering a single facet (e.g., URIs, labels, instances of an entity, etc.), covering all of the facets or just a subset thereof. Typically, a distinction is made between the way ontology entities are named and the way these are related to other entities within the ontology, the former being termed depending on the context "lexical", "linguistic", "terminological", etc. while the latter is usually qualified as "structural". Structural similarity measurement is performed as a form of graph matching whereas lexical one boils down to either string comparison or matching of representations of the semantic of terms in entity names.

Our matching tool OLA [1, 2] targets OWL-DL (formerly OWL-LITE) ontologies. It applies an integrated approach towards similarity, i.e., calculation of a unique score for all entity pairs based on an aggregation of all facet-wise comparison results (facets here stand for the relationships between OWL entities). Computation is *exhaustive* on entity descriptions meaning that all facets are covered. The similarity is defined through a category-sensitive[3] yet universal operator that basically computes a linear combination of facet similarities. As facets mostly represent entities of their own, the similarity definition gets circular and hence cannot be directly computed. OLA considers such definitions as equations to solve and approximates their solutions through an iterative fixed-point-bound process. As initial values are based on name comparison while iterations basically perform similarity exchange between pairs of neighbor entities, OLA similarity is a trade-off between the aforementioned structural and lexical aspects.

Previous participation of OLA in the alignment competitions [3, 4], despite globally positive outcome, have put the emphasis on a certain lack of homogeneity among the computational mechanisms at different levels of the similarity model that harm efficiency. These were traced back to the somewhat overloaded structure of the OL-graphs, the graph-based representation of OWL ontologies that was used to support the similarity computation.

Restoring homogeneity and improving efficiency was the motivation behind the OLA2 version that is developed jointly by UQÀM and INRIA. It introduced a flattened version of the OL-graph model where at most one scalar value is admitted in vertices while all the remaining information is in the edges. This allowed the iterative re-calculation of the similarity scores to be modeled as matrix operation without losing the valuable properties of the result nor the process.

To that end, the ontology graphs are combined into a product-like construct, the *match graph*, where vertices and edges are products of counterpart elements from the ontology graphs. Similarity computation represents an iterative value propagation across the match graph starting with initial values yielded by name comparison. The innovation is the matrix product used in re-calculations: the adjacency matrix of the match graph is used as the multiplicative factor leading to a fixed point. The resulting method is a step further towards structure-domnated similarity computation as it encompasses all relationships of an ontology entity whereas the previous version tended to disregard non-descriptive relationships (e.g., the one between a OWL class and a relation whose range the class represents).

Our new method has outperformed the initial version of OLA on both competition tests (benchmark) and efficiency, although many of the optimizations from previous years have not been implemented in it. More dramatically, its modular implementation eases future improvements.

---

[3] Entity categories, e.g., OWL class, property, object, data type, value, etc., compare to meta-classes of language meta-model

2 System Overview

OLA is an open-source tool, implementing the OLA algorithm (for *O*WL-*L*ite *A*lignment [2]), jointly developed by teams at University of Québec at Montréal and INRIA Rhône Alpes.

2.1 General purpose statement

The primary goal behind the OLA tool design is to perform alignment of ontologies expressed in OWL [1, 2], with an emphasis on OWL-DL (formerly OWL-LITE). The system offers similarity-based alignment on graph-like ontology representations. Beside alignment, it features a set of auxiliary services supporting the manipulation of alignment results.

2.2 Ontology graph model

Traditionally, an ontology is viewed as a set of *entities* and a set of *relationships* between those entities. This view underlies the translation of the ontologies into a graphs structure where entities become vertices and relationships edges. Yet in the new settings, beside language entity categories such as *classes*, *objects*, *relations*, *properties*, *property instances*, and *data types*, *data values*, less traditional ones are considered, i.e., *tokens* (including comments on entities) and *cardinalities*. The list of relationships is accordingly completed: together with previously existing in the OL-graph format *subsumption*, *instantiation*, *attribution*, *domain*, *range*, *restriction*, *valuation*, and *all* relationships, we exploit *card* and *name*. Fig. 1 provides an overview of the meta-model for the ontology graph format.

Both vertices and edges in the graph are labeled by their respective entity category/relationship.

As a support for similarity computation, the product of both ontology graphs, or their *match* graph, is composed. The vertices of the match graphs are pairs of vertices from opposite ontology graphs. In its basic version, the match graphs comprises the cartesian product of both vertex sets, i.e., same category is not required for vertices v_1 and v_2 to form a product vertex. Clearly, product vertices correspond to the variables of the equation system in the previous version of OLA. They embed a "weight" value which stands for the similarity and is computed iteratively (see below).

In contrast, match graph edges require strict correspondence: An edge labeled l exists between compound vertices (v_1, v_2) and (v_1', v_2') iff there exist an edge labeled l between v_1 and v_1' in the first graph another one between v_2 and v_2' in the second graph. Edges in the match graph are also weighted yet their weights are effective as they correspond to the weights on neighboring sets in the OL-graphs in version one.

Similarity model Similarity between entities of the initial graphs is reflected by the weight or importance index of the corresponding match graph node. The underlying computational model is the value propagation as described in [5] (and used in a range of alignment methods starting with [6]) across the graph. We recall that, the adjacency

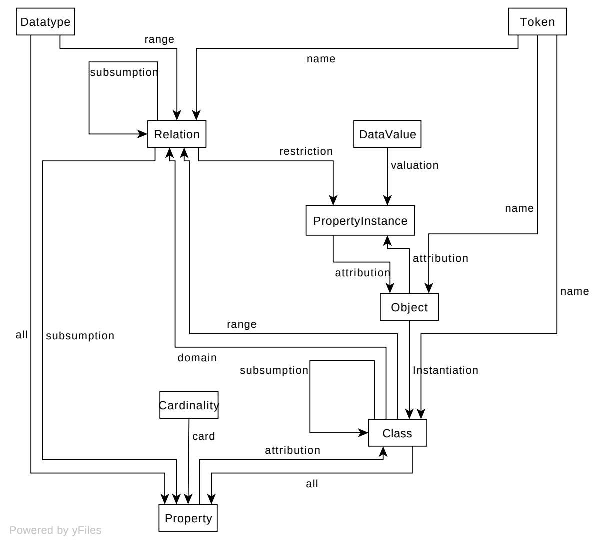

Fig. 1. Relationships between ontology entities with respect to the cluster to which they belong

matrices of the initial graphs are used to produce a larger matrix M reflecting both the inbound and the outbound neighbors of a match vertex. M is then used as a multiplier for the similarity vector V. Thus, starting with initial values, typically 1 for all entity pairs, V evolves according to the following iterative dependency: $V_k = M \times V_{k-1}$ ($k = 1, 2, 3 ...$), until a fixed point is reached.

Yet the model has been adapted – and even somewhat spoiled – as to compute the similarity defined in the reference OLA version. Recall that for a category X together with the set of relationships it is involved in, $\mathcal{N}(X)$, the similarity measure $Sim_X : X^2 \rightarrow [0, 1]$ is defined as follows:

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

The function is normalized, i.e., the weights $\pi^X_{\mathcal{F}}$ sum to one, $\sum_{\mathcal{F} \in \mathcal{N}(X)} \pi^X_{\mathcal{F}} = 1$. Moreover, the set functions $MSim_Y$ compare two sets of nodes of the same category and extract a maximal pairing thereof that further optimizes the total similarity (see [2] for details).

In order to simulate the above family of functions, the graph-based model introduces first-class weights on relationship sets adjacent to a match vertex. Hence the adjacency matrix of the match graph that is central to the value propagation is not purely Boolean: values between 0 and 1 appear.

Moreover, the above equation is modified to reflect the evolving contribution of neighbor nodes:

$$V_k = M_k \times V_{k-1} \text{ } (k = 1, 2, 3 ... \text{ and } M_k = f(M_{k-1}, V_{k-1}))$$

Here M_k reflects possible changes in pairings between contributing vertices within a set of neighbors, a recalculation that is done at each iteration in basic OLA. Hence f involves the current solution vector, as well.

Further adaptation of the original method is the initialization of V_0 with the results from name comparison for entities in a match vertex. This corresponds strictly to the initialization of the equation system in OLA.

The last adaptation completely changes the ideology of the method as it plays against the very basic principle of propagation: at each step a value at a vertex is replaced by a combination of the values of all its neighbors. Our understanding is that this is a major reason for the convergence of the computed values only for even steps.

In our model, the stabilizing role of name similarity (which is computed only once) has been secured by a representation trick. In fact, the vertices representing the sources of stable similarity, i.e., *token*, *cardinality*, *data type*, etc. are provided with a local looping edge while fixing their weights till the end of the process. Hence the respective OWL entity node that is identified by the token gets the same value at each iteration.

The above process provenly converges towards a solution vector V_∞.

Past optimizations A number of optimizations have been implemented within the system mainly aimed at making the weights – matching – similarity scheme more flexible. First, mechanisms for weight adaptation, both at entity and ontology level have been designed. The goal is to insure that the absence of a specific facet locally, i.e., for an entity pair, or globally, i.e., for all pairs, does not result in unbalanced similarity scores. An extension thereof based on simple statistics provides the basis for an even further adaptation of initial facet weights that in a way reflective of the relative importance of each facet. The nature of the name measure to use, i.e., string-based or term-based, is heuristically determined based on similar reasoning.

Many of the optimizations could not be implemented in the current version. Yet a new optimization could be designed to help offset the impact of meaningless names. It consists in replacing the label of a token vertex with the set of labeled paths that head towards that vertex.

Link to OLA:
`https://gforge.inria.fr/frs/?group_id=271`

Link to alignments and parameters file :
`http://ola.gforge.inria.fr/results/OAEI-2007-OLA.zip`

3 Results of Execution on Test Cases

3.1 Benchmarks

#101-104:

– Language variations

- **Mean Precision =** 1.00 and **Mean Recall =** 1.00

#201-204:

- Alteration of names and/or suppresion of comments
- **Mean Precision =** 0.92 and **Mean Recall =** 0.92

#205-210:

- Synonyms and/or foreign language
- **Mean Precision =** 0.90 and **Mean Recall =** 0.90

#221-223:

- Alteration of specialization hierarchy
- **Mean Precision =** 0.99 and **Mean Recall =** 1.00

#224-228:

- Absence of instances, properties and/or restrictions
- **Mean Precision =** 1.00 and **Mean Recall =** 1.00

#230-231:

- Classes expanded or flattened
- **Mean Precision =** 0.96 and **Mean Recall =** 1.00

#232-247:

- Alteration/suppresion of specialization hierarchy
- Suppression of some properties and/or instances
- **Mean Precision =** 0.97 and **Mean Recall =** 1.00

#248-266:

- Alteration of all names
- Suppression of all comments
- Alteration of specialization hierarchy (in most cases)
- Suppression of some instances and/or properties
- **Mean Precision =** 0.77 and **Mean Recall =** 0.51

#301-304:

- Real-world ontologies
- **Mean Precision =** 0.63 and **Mean Recall =** 0.73

3.2 Conference

- We align every possible couple of ontologies
- We ran OLA2 with the set of parameters used for the benchmarks test case

3.3 Directory

We ran OLA2 with the set of parameters used for the benchmarks test case.

3.4 Others

OLA2 was unable to run because of the large size of matrices extracted from ontology graphs.

4 General Comments

4.1 Results

OLA2 significantly improves on the previous version of OLA. This may be seen on the benchmark results which are better. The additional benefits of the new implementation are extensibility and modularity of code.

Yet the method heavily relies on similarity of vertice labels (entity names or paths). A look on the tests where OLA scored poorly reveals that.

- #201-204 & #248-266: choice among entities having similar roles within their respective ontology graphs (e.g., test #253),
- #205-210 & #301-304: lack of semantic string distance and language translator,
- #221-223 & #230-231 & #232-247: no explicit inheritance edge between classes and properties

4.2 Future Improvements

1. Factorization of ontology graphs to run OLA on large-size ontologies [7, 8];
2. Search for the set of optimal of weight values;
3. Integration of semantic string distances [9] within the OLA matching process;
4. Integration of explicit inheritance edges among classes and among properties.

5 Conclusion

OLA2 is arguably better now than two years ago. The progress on real-world ontologies (30X), a class the previous version had difficulties dealing with, is encouraging. Yet more encouraging is the fact that these results have been obtained with very few adaptation tricks. In this respect, our next target will be the weight computing mechanisms of previous OLA.

Acknowledgments

This work was partly supported by the NSERG Discovery grant of the third author. The second author has been supported by the European integrated project NeOn (IST-2005-027595) and the RNTL project Web-Content. A short term visit by the first author at EXMO research team, INRIA Rhône-Alpes, France was jointly supported by the NSERG and NeOn grants.

References

1. Euzenat, J., Valtchev, P.: An integrative proximity measure for ontology alignment. In: Semantic Integration Workshop, Second International Semantic Web Conference (ISWC-03). (2003)
2. Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in OWL-Lite. In: European Conference on Artificial Intelligence ECAI-04. (2004) 333–337
3. Euzenat, J., Loup, D., Touzani, M., Valtchev, P.: Ontology alignment with OLA. In Sure, Y., Corcho, O., Euzenat, J., Hughes, T., eds.: Proceedings of the 3rd ISWC'2004 workshop on Evaluation of Ontology-based tools (EON). (2004) 59–68
4. Euzenat, J., Guégan, P., Valtchev, P.: OLA in the OAEI 2005 alignment contest. In Ashpole, B., Euzenat, J., Ehrig, M., Stuckenschmidt, H., eds.: Proceedings of the K-CAP Workshop on Integrating Ontologies 2005. (2005) 61–71
5. Blondel, V.D., Heymans, A.G.M., Senellart, P., Dooren, P.V.: A measure of similarity between graph vertices. with applications to synonym extraction and web searching. SIAM Review **46** (2004) 647–666
6. Hu, W., Jian, N., Qu, Y., Wang, Y.: GMO: A graph matching for ontologies. In: K-CAP Workshop on Integrating Ontologies. (2005) 43–50
7. Papadopoulos, C., Voglis, C.: Drawing graphs using modular decomposition. In Healy, P., Nikolov, N.S., eds.: Graph Drawing. Volume LNCS 3843., Springer (2005) 343–354
8. Ryan, M.R.K.: Graph decomposition for efficient multi-robot path planning. In: IJCAI. (2007) 2003–2008
9. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)