

OACAS: results for OAEI 2011

Sami ZGHAL¹, Marouen KACHROUDI¹, Sadok BEN YAHIA¹, and Engelbert MEPHU NGUIFO²

¹ University of Tunis El Manar

Computer Science Department, Faculty of Sciences of Tunis, Tunisia

Campus Universitaire, 1060 Tunis, Tunisia

{sadok.benyahia, marouen.kachroudi}@fst.rnu.tn

sami.zghal@planet.tn

² LIMOS CNRS UMR 6158, Complexe scientifique des Cézeaux

BP 125, 63173 Aubiere Cedex, France

mephu@isima.fr

Abstract. Ontologies are the kernel of semantic Web. They allow the explicitation of the semantic purpose for structuring different fields of interest. In order to harmonize them and to guarantee the interoperability between these resources, the topic of alignment of ontologies has emerged as an important process to reduce their heterogeneity and improve their exploitation. The paper introduces a new method of alignment of OWL-DL ontologies, using a combination and aggregation of similarity measures. Both ontologies are transformed into a graph which describes their information. The proposed method operates in two steps: local (linguistic similarity composition and neighborhood similarity) step and the aggregation one.

1 Presentation of the system

The method, OACAS [1] (Ontologies Alignment using Composition and Aggregation of Similarities), introduces an alignment algorithm of OWL-DL (Ontology Web Language Description Logic) ontologies. The main thrust of this method is the application of the most suitable similarity measure depending of the category of the node in the ontology. In addition, the OACAS method explores a wider neighborhood than do the pioneering methods of the literature. Carried out experiments showed that OACAS presents very encouraging values of the commonly used evaluation metrics for the assessment of ontologies alignment.

1.1 Specific techniques used

The proposed method, OACAS, alignes two ontologies. Both ontologies are described in the OWL-DL language [2]. Both ontologies are transformed in two graphs O-GRAPHS. The obtained graphs are parsed in order to produce the alignment process out.

Mapping of an OWL-DL Ontology to an O-GRAFH. The process of building the graphs allows to faithfully map the considered ontologies to be aligned in two graphs, *called* O-GRAFHs. An O-GRAFH describes all the information categories included in an OWL-DL ontology: classes, relations and instances. Both classes and instances represent the nodes of the graph. The relations between these different entities are induced by the links of an O-GRAFH. Each entity of the ontology is formalized through an associated notion to the RDF formalism [3]. OWL-DL ontology entities are described thanks to OWL language constructors. These constructors are represented through RDF triplets: <subject, predicate, object>. In an OWL-DL ontology, a class or a relation description is an RDF triplet. The subject corresponds to the class or to the relation. Predicates are OWL primitives, which are OWL and RDF properties. Each property, used in a triplet, sketches a knowledge of the described entity. The arrangement of those nuggets of knowledge constitutes the entity definition. The representation of an OWL-DL ontology through an O-GRAFH permits to load the ontology in main memory only once. An O-GRAFH, stored in main memory, statistically reduces the time required to access initial OWL-DL ontology disk resident file.

The alignment method. The introduced OACAS method lays on a composition and an aggregation of similarity computation based model. The method starts by exploring the O-GRAFH structure. It determines the nodes of both ontologies to be aligned and gets out the similarity measures. For each node of the same category (or cluster), the alignment model computes similarity measures between descriptors by using appropriate functions. Thus, this function considers all the descriptive information of this couple (name, comment and label) as well as its neighborhood structure. An aggregation function combines the similarity measures and the node's structures of the nodes to be aligned. The algorithm implementing the OACAS method takes as input two OWL-DL ontologies to be aligned and produces an RDF file containing the aligned nodes as well as their similarity measures. The alignment method operates into two successive steps. The first one computes the local similarity, whereas the second one computes the aggregation similarity.

First step: Local similarity

The local similarity computation is performed into two successive stages. The first one computes many linguistic similarity measures and aggregates them for each couple of nodes belonging to the same category (or type). The second one computes neighborhood similarities by exploiting the structures of the nodes to be aligned.

The linguistic similarity computation is carried out once for each node of the same cluster (node of the same type) in the beginning of the alignment process. The linguistic similarity measures of couples of entities of the same type (class, property and instance) are computed. The names of properties and instances are used to compute linguistic similarities. For class category, the computation of the linguistic similarity considers both the comments and labels. The computation of linguistic similarities uses different similarity measures. Those measures are adapted to different descriptors (names, comments and labels) of the entities to be aligned. Different similarity values obtained, for the descriptors, are composed. This composition assigns weights to each similarity

measure of descriptors. The sum of the assigned weights to different similarity values is equal to 1. This unit sum guarantees that the composition of the similarity produces a normalized value (between 0 and 1). The LEVENSSTEIN similarity measure [4] is used to compute the similarity value between the names of ontological entities. The Q-GRAM similarity measure [5] computes the similarity value between the comments of the ontological entities. The JARO-WINKLER similarity measure [6] computes the similarity value between the labels of ontological entities. The LINGUISTIC function computes composed linguistic similarity of couples of nodes of both ontologies to be aligned, *i.e.*, O_1 and O_2 . It takes as input *(i)* both ontologies sketched by two corresponding O-GRAPHS; *(ii)* linguistics similarity functions (*i.e.*, $Funct$); and *(iii)* weighted attributed to the descriptors nodes (*i.e.*, Π_D). As a result, it produces a composed linguistic similarity vector, V_{CLS} , for each couple of n nodes. The similarity function $Funct$ considers two nodes, N_1 and N_2 , and returns the linguistic similarity value of the descriptor, Sim_{LD} . LEVENSSTEIN or Q-GRAM or JARO-WINKLER implements the similarity function, $Funct$, depending of the type of the nodes. Composed linguistic similarity, Sim_{CL} , is computed depending of the descriptors of nodes to be aligned and associate weights to each descriptor, Π_D . Both nodes (N_1 and N_2) and the associated composed linguistic similarity (Sim_{CL}) are added to the composed linguistic similarity vector (V_{CLS}). The composed linguistic similarity of different couples of entities will be used to compute the neighborhood similarity as sketched in the following.

The neighborhood similarity considers both ontologies to be aligned (*i.e.*, O_1 and O_2), the composed similarity vector (V_{CLS}), the weights assigned to each category (Π_O) and the weights associated to the neighbor level (Π_L). Therefore, it produces the neighborhood similarity vector, V_{NS} . The neighborhood similarity computation needs composed linguistic similarity of the couple of nodes to be aligned and the nodes structures. Neighborhood nodes are organized by category, node having the same type. The neighborhood similarity computation propagates similarity into two successive neighborhood levels. The first level (level 1) includes direct neighbors of the nodes to be aligned whereas second one (level 2) contains indirect neighbors. Direct neighbors of the first level represent nodes having direct relationship with the node under consideration. Neighbors of the second level represent nodes having relationship with the nodes of the first one. The neighbors entities of the first level are clustered into three categories (classes, instances or properties). Each category (or cluster) includes ontological entities having the same type. After the step of clustering, the neighborhood similarity is computed between those categories. The neighborhood nodes of the level 2 are treated in the same manner as the neighbors of the first one. The neighborhood similarity by group $MSim$ takes nodes from vectors VN_1 and VN_2 regrouped by category (where VN_1 and VN_2 denote a vector nodes of O_1 and O_2). The process computation uses the "Match-Based similarity" [7] as follows:

$$MSim(E, E') = \frac{\sum_{(i,i') \in Pairs(E, E')} Sim_{CLS}(i, i')}{Max(|E|, |E'|)}. \quad (1)$$

Both sets E and E' represent nodes of the same cluster belonging respectively to vectors VN_1 and VN_2 . The neighborhood similarity, Sim_N , is computed using Equation 2:

$$Sim_N = \sum_{i \in \{1, 2\}} (\Pi_{Vi} (\sum_{(E, E')} \Pi_{(E, E')} MSim(E, E'))), \quad (2)$$

where i stands for the level (*i.e.*, 1 or 2). The neighborhood similarity, Sim_N is a normalized value, since the sum of weights assigned to different neighbors is equal to 1, ($\Pi_{V1} + \Pi_{V2} = 1$). Direct neighbors (level 1) have more important relationships than those of indirect one (level 2). Thus, nodes of level 1 have an important impact on the produced alignment. For this reason, the weight assigned to the first level, $\Pi_{V1} = 0.8$, is more important than the one assigned to the second level, $\Pi_{V2} = 0.2$. In addition, the sum of weights assigned to the category of nodes is equal to 1 ($\sum(\Pi_C) = 1$). Those weights are uniformly assigned between the different categories. The neighborhood similarity is computed thanks to an iterative process, level by level. The obtained values of the composed linguistic similarity, *i.e.* V_{CLS} , and neighbors similarity, *i.e.* V_{NS} , are combined in order to compute aggregation similarity.

Second step: Aggregation similarity

The aggregation similarity is a combined similarity between the local similarities (the composed linguistic similarity and the neighborhood similarity). Function AGGREGATION needs to have in input both ontologies to be aligned, O_1 and O_2 , the two similarity vectors, V_{CLS} and V_{NS} , and the weights attributed to the both kind of similarities, Π_{CL} and Π_N . It produces the aggregated similarity vector, V_{AS} . For each couple of entities, N_1 and N_2 , of the same category of the both ontologies to be aligned, O_1 and O_2 , the aggregated similarity is computed as follows:

$$Sim_A(e_1, e_2) = \Pi_{CL} Sim_{CL}(e_1, e_2) + \Pi_N Sim_N(e_1, e_2). \quad (3)$$

Note that the sum of the weights, attributed to each kind of similarity, is equal to 1 in order to have a normalized aggregation (between 0 and 1). In addition, the sum of weights is equal to 1 ($\Pi_{CL} + \Pi_N = 1$). In the next section, we focus on the experimental evaluation of OACAS.

1.2 Adaptations made for the evaluation

The main objective of the adaptations with the OACAS method is to find the best combination of linguistic measures. In the experimental study, various measures have been used. The goal is to experiment different measures in order to find the more appropriate measure associated to the node descriptors. In order to achieve the objective, 27 arrangements of tests have been experimented. Each test uses a particular combination of similarity measures to compute linguistic similarities between the descriptors of entities to be aligned. During the process of the carried out tests, different weights were assigned to the descriptors (names, comments and labels). The nodes to be aligned can have different descriptors. Depending on the descriptors of the nodes, different

weights are attributed. In the case where the nodes are described by three descriptors, the weights are 0.8, 0.1 and 0.1 associated respectively to the names, comments and labels. Whereas the nodes contain only names and comments descriptors, the weights are respectively 0.85 and 0.15. The weights 0.85 and 0.15 are assigned to the names and labels where those the entities are described by them. The experimental results obtained are developed in the next subsection.

The combination using three different linguistic similarities (LEVENSHTEIN, Q-GRAM and JARO-WINKLER) is the best one. In fact, the LEVENSHTEIN measure is more appropriate for computing linguistic similarity between the names of entities to be aligned. Whereas, the Q-GRAM measure is more indicated to compute linguistic similarity between comments of ontological entities. JARO-WINKLER measure is more appropriated for computing linguistic similarity between the labels of entities to be aligned. Indeed, names and labels of ontological entities are short strings. For this type of strings, LEVENSHTEIN and JARO-WINKLER measures are more adapted to compute the linguistic similarity. Comments are strings composed with many words. For this type strings, the Q-GRAM measure gives the best linguistic similarity values.

2 Results

In this section we present the results obtained by OACAS method. Our method produces result for the benchmark tests sets and conference track.

2.1 Benchmark

The benchmark tests sets can be divided into eight groups: 10x, 20x, 22x, 23x, 24x, 25x, 26x and 30x. For each group the mean values of precision and recall are computed. Table 1 shows the values of the evaluation metrics.

Test	Precision	Recall
10x	0.71	1.00
20x	0.44	0.48
22x	0.64	1.00
23x	0.57	1.00
24x	0.40	0.50
25x	0.34	0.46
26x	0.17	0.40
30x	0.47	0.61

Table 1. Mean values of precision and recall for each group of tests

For the group of tests 10x, OACAS achieves precision and recall values of 71% and 100% respectively. Since the ontologies in those tests have complete information, which can be used for alignment. The precision mean value can be explained by the fact that OACAS produces alignment containing individuals correspondences. Those correspondences are not included in the reference alignments.

The OACAS method obtains degraded mean values of precision and recall for the family of tests 20x. This degradation can be interpreted by the fact that the ontologies to be aligned contain translated or synonyms descriptor of entities. Our method relies on syntactical treatment of ontological entities.

For the groups of tests 22x and 23x, OACAS obtains 64% and 57% of precision mean values respectively and 100% of recall. The origin of those results is the absence of proprieties, individuals and a flattened hierarchy.

For the tests 25x and 26x combine linguistic and structural problems. For this reason OACAS method provides low mean values of precision and recall.

The problem of individuals absence is still the main handicaps in the real case tests 30x.

2.2 Conference

Table 2 shows the precision and recall values obtained for each test of Conference track.

Test	Precision	Recall
cmt-confOf	0.07	0.40
cmt-conference	0.04	0.29
cmt-edas	0.08	0.67
cmt-ekaw	0.04	0.42
cmt-iasted	0.03	0.95
cmt-sigkdd	0.10	0.79
confOf-edas	0.10	0.55
confOf-ekaw	0.12	0.50
confOf-iasted	0.04	0.44
confOf-sigkdd	0.05	0.52
conference-confOf	0.06	0.46
conference-edas	0.06	0.52
conference-ekaw	0.14	0.68
conference-iasted	0.03	0.33
conference-sigkdd	0.08	0.53
edas-ekaw	0.08	0.52
edas-iasted	0.05	0.52
edas-sigkdd	0.08	0.57
ekaw-iasted	0.04	0.57
ekaw-sigkdd	0.07	0.60
iasted-sigkdd	0.10	0.81

Table 2. Values of precision and recall for Conference track

3 General comments

We participate this year for the first time in OAEI and see the result obtained by our method. The evaluation and comparison of ontology alignment and schema matching components as OAEI is very useful for the development of such technologies.

4 Conclusion

In this paper, we introduced an alignment method of OWL-DL ontologies. The new proposed method OACAS, allows to exploit at most the informative present within in an ontology described in OWL-DL. The process of alignment in the OACAS method, contains two phases: a local phase and a phase of aggregation. The local phase allows to calculate the linguistic similarity consisted as well as the neighborhood similarity. This two similarities are combined during the second phase to determine the aggregation similarity.

References

1. Zghal, S., Kachroudi, M., Ben Yahia, S., Mephu Nguifo, E.: OACAS: Ontologies alignment using composition and aggregation of similarities. In: Proceedings of the 1st International Conference on Knowledge Engineering and Ontology Development (KEOD 2009), Madeira, Portugal (2009) 233–238
2. Smith, M.K., Welty, C., McGuinness, D.L.: OWL: Ontology Web Language Guide. Technical report, W3C: World Wide Web Consortium, <http://www.w3.org/TR/2004/REC-owl-guide-20040210/> (February 2004)
3. Klyne, G., Carroll, J.J.: Resource Description Framework (RDF): Concepts and Abstract Syntax. Technical report, W3C: World Wide Web Consortium, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/> (February 2004)
4. Levenshtein, I.V.: Binary codes capables of corrections, deletions, insertions and reversals. Soviet Physics-Doklady **10**(8) (1966) 707–710
5. Ukkonen, E.: Approximate string-matching with q-grams and maximal matches. Theoretical Computer Science **92**(1) (1992) 191–211
6. Winkler, W.: The state of record linkage and current research problems. Technical Report 99/04, Statistics of Income Division, Internal Revenue Service Publication (1999)
7. Valtchev, P.: Construction automatique de taxonomies pour l'aide la représentation de connaissance par objets. Thèse de doctorat, Université de Grenoble 1, France (1999)