SODA: an OWL-DL based ontology matching system

Sami Zghal[1,2], Sadok Ben Yahia[2], Engelbert Mephu Nguifo[1], and Yahya Slimani[2]

[1] CRIL CNRS FRE 2499, Artois University, IUT of Lens
Rue de l'Université - S.P. 16, 62307 Lens Cedex, France
{zghal, mephu}@cril.univ-artois.fr
[2] Computer Science Department, Faculty of Sciences of Tunis, Tunisia
Campus Universitaire, 1060 Tunis, Tunisia
{sadok.benyahia, yahya.slimani}@fst.rnu.tn

Abstract. This paper describes SODA a novel ontology alignment method for the OWL-DL format. SODA uses a new approach that consists in computing local and semantic similarities among ontological elements.

1 Presentation of the system

SODA [1] (Structural Ontology OWL-DL [2] Alignment), is a new approach that aligns two OWL-DL ontologies using similarity measures [3]. Both OWL-DL ontologies are transformed in two corresponding graphs DL-GRAPH which describe all information in the ontologies. SODA uses the DL-GRAPH to align the two ontologies. It operates into successive steps. The first step, computes local similarity by means of linguistic and structural similarities, whereas the second one computes the semantic similarity. Figure 1 depicts the architecture of SODA system.

1.1 Specific techniques used

Each OWL-DL ontology to be aligned is transformed into a non oriented graph called DL-GRAPH. All the information belonging to OWL-DL ontology are faithfully mapped into the DL-GRAPH. Nodes of the proposed graph represent classes, properties and instances. The DL-GRAPH nodes represent six types (named also categories) of entities that may exist in an OWL-DL ontology: *i.e.*, concepts, instances of concepts, data types, values of data types and class properties (object nature and data type nature). Connections between the graph nodes map the relationships between the entities in an OWL-DL ontology. It is worthily noted that an OWL-GRAPH describes all the semantic relations between different entities of an ontology. A graph DL-GRAPH allows to represent four kinds of links: *specialization, attribution, instantiation* and *equivalence*. DL-GRAPHS are exploited by the alignment model SODA. Similarity measures are used to compare the components of the graphs in order to obtain the correspondence between them. Nodes and links of the two graphs are compared to get out the correspondence between different ontological entities using similarity measures. The output algorithm is an RDF file containing all the correspondences between the entities and the similarity measure values.

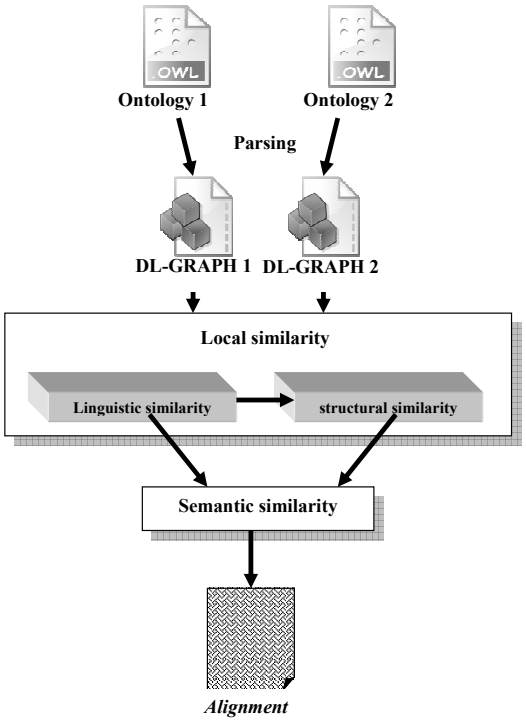

Fig. 1. Architecture of SODA

SODA explores the structure of DL-GRAPH to compute the similarity values between the nodes of both ontologies. The alignment model associates to each category of nodes an aggregation function. This function takes in consideration all the similarity measures and the structure of couple of nodes to be matched. This aggregation function explores all descriptive information of nodes.

SODA operates into two successive steps: local and semantic. The first step, implemented via PHASE1_LINGSIM (see Algorithm 1) and PHASE2_STRUCTSIM (see Algorithm 2) functions, computes the local similarity (linguistic and structural one). The second step, *c.f.* the PHASE3_SEMSIM function (see Algorithm 3), computes the semantic similarity. Table 1 summarizes the notations that are used in the description of our algorithms.

Local similarity

The computation of the local similarity is carried out in two phases. The first phase allows to compute the linguistic similarity for each couple of node of the same category. The second phase allows to compute the structural similarity using the structure of neighbors of the nodes to be aligned.

- O_1,O_2: two OWL-DL ontologies for alignment - V_{LS}: linguistic similarity vector - V_{SS}: structural similarity vector - V_{VSEMS}: semantic similarity vector Each node of the ontology is characterized by: - **Type**: node type - **Name**: node name
Each element of the vectors V_{SL}, V_{SS} et V_{VSEM} is characterized by: - **Node 1**: the node of ontology O_1 - **Node 2**: the node of ontology O_2 - **Sim**: the similarity value

Table 1. Algorithm notations

Algorithm 1 (*c.f.*, PHASE1_LINGSIM function) computes the linguistic similarity measure. The name of properties and instances are used to compute linguistic similarity. For classes, the computation of linguistic similarity integrates also comments and labels. The computation of linguistic similarity is done only once for each node of the same category. JARO-WINKLER or MONGE-ELKAN [4] functions are used to compute the linguistic similarity. JARO-WINKLER measure is more adapted for short strings, like those representing names and labels [4]. Besides, MONGE-ELKAN measure is better indicated for long strings, *e.g.* the comments [4]. PHASE1_LINGSIM function computes the linguistic similarity of couple of nodes of both considered ontologies. PHASE1_LINGSIM function takes as input the two ontologies O_1 and O_2 and the linguistic similarity function $Funct_{LS}$. COMPUTELINGSIM function (*c.f.*, line 8 of Algorithm 1) takes as an input two nodes, $Node_1$ et $Node_2$, and linguistic similarity function. PHASE1_LINGSIM function returns as an output linguistic similarity value Sim_L. This function implements the JARO-WINKLER or the MONGE-ELKAN measures. Linguistic similarity of the different couples of nodes are used after that in the computation of the structural similarity.

Structural similarity is computed by using linguistic similarity of the couple of nodes to align and the neighborhood structure. Adjacent neighbor nodes of the entities are grouped by category, *c.f.* PHASE2_STRUCTSIM. This function takes as input two ontologies O_1 and O_2 to align, linguistic similarity vector V_{LS} and weights associated for each category Π_C. EXTRACTNODES function, (*c.f.*, lines 9 - 11 of Algorithm 2), allows to extract for each node, its neighbors and to put them in V_{Node_i}, where $Node_i$ is a node of O_1 or O_2. V_{Node1} and V_{Node_2} vectors and weights associated for each category, Π_C, are used by the COMPUTESTRUCTSIM function (*c.f.*, line 13 of Algorithm 2) to compute the structural similarity, Sim_S. To work out, the following "***Match-Based similarity***" [5, 6] is used to compute similarity between two categories (one in the first ontology and the other in second one):

$$MSim(E, E') = \frac{\sum_{(i,i') \in Paires(E,E')} Sim(i, i')}{Max(|E|, |E'|)},$$

```
1  Function : PHASE1_LINGSIM
   Data:
     1. O1 and O2 : two ontologies to align
     2. Funct_LS : linguistic similarity function

   Results: V_LS : linguistic similarity vector
2  Begin
3    /*Parse all nodes of the ontology O1*/
4    forall (Node1 ∈ O1) do
5      /*Parse all nodes of the ontology O2*/
6      forall (Node2 ∈ O2) do
7        If Node1.type=Node2.type then
8          Sim_L = COMPUTELINGSIM(Node1,Node2)
9          /*Add Node1,Node2 and Sim_L to V_LS*/
10         Add((Node1,Node2,Sim_L),V_LS)
11   return(V_SL)
12 End
```

Algorithm 1: PHASE1_LINGSIM

where E et E' represent two sets of nodes belonging to the same category in O_1 and O_2. This function uses the local similarities of the couple (i, i') already computed. Structural similarity is computed by aggregating the "*Match-Based similarity*" of each group of adjacent neighborhood nodes by category. A weight is attributed for each group to have a normalized structural similarity. Each category has the same weight which is equal to 1 over the number of groups (categories). Structural similarity, Sim_S, is computed as follows:

$$Sim_S = \sum_{(E,E') \in (V_{Node_1}, V_{Node_2})} \Pi_{(E,E')} MSim(E, E').$$

The structural similarity, Sim_S, is normalized since $\sum(\Pi_{(E,E')}) = 1$. Values of the linguistic similarity vector, *i.e.* V_{LS}, and structural similarity vector, *i.e.* V_{SS}, already obtained are combined to compute the semantic similarity.

Semantic similarity

The semantic similarity is a combined similarity measure of linguistic and structural similarities (local similarity). Algorithm 3 takes as input the two ontologies to be aligned, the two similarity vectors (the linguistic one, V_{LS}, and the structural one, V_{SS}). It takes also the weights attributed to the linguistic similarity and structural similarity (Π_L and Π_S =). Optimal weight values can be determined through several experiments. Algorithm 3 outputs semantic similarity vector, V_{SEMS}. EXTRACTSIM function (*c.f.*, lines 9 - 11 of Algorithm 3) extracts corresponding values of similarities of the nodes ($Node_1$ or $Node_2$) from similarity vectors (V_{LS} or V_{SS}). Each couple of entities, $Node_1$ and $Node_2$, of the same category, the semantic similarity is computed as follows (*c.f.*, line 13 of Algorithm 3):

$$Sim_{SEM}(e_1, e_2) = \Pi_L Sim_L(e_1, e_2) + \Pi_S Sim_S(e_1, e_2).$$

1.2 Link to the system and parameters file

At the following URL the system SODA can be downloaded: www.cril.univ-artois.fr/~mephu/OAEI2007/systemSODA.rar.

Function : PHASE2_STRUCTSIM
Data:
1. O_1 et O_2: two ontologies to align
2. V_{LS}: linguistic similarity vector
3. Π_C: weights for each category of node

Results: V_{SS} : structural similarity vector
Begin
  /*Parse all nodes of the ontology O_1*/
  forall *(*$Node_1 \in O_1$*)* **do**
    /*Parse the nodes of the ontology O_2*/
    forall *(*$Node_2 \in O_2$*)* **do**
      If $Node_1.\texttt{type}==Node_2.\texttt{type}$ **then**
        /*Extract form V_{Node1} neighbor nodes $Node_1$*/
        V_{Node_1}=EXTRACTNODE($Node_1$)
        /*Extract form V_{Node2} neighbor nodes $Node_2$*/
        V_{Node_2}=EXTRACTNODE($Node_2$)
        /*Compute the structural similarity*/
        Sim_S = COMPUTESTRUCTSIM(V_{Node_1},V_{Node_2},Π_C)
        /*Add $Node_1$, $Node_2$ and Sim_S to V_{SS}*/
        ADD(($Node_1$,$Node_2$,Sim_S),V_{SS})
  return(V_{SS})
End

Algorithm 2: PHASE2_STRUCTSIM

1.3 Link to the set of provided alignments

At the following URL the contest results of SODA are available: www.cril.univ-artois.fr/~mephu/OAEI2007/resultsSODA.rar. As the evaluation process of OAEI 2007 Challenge was over, it is not possible to comment those results here. Some details of results can be found in [1], especially for the benchmark dataset.

2 Conclusion

In this paper, a new ontology alignment system for OWL-DL format is described. Results obtained on OAEI 2007 benchmarks and directory are available on website. Evaluation of benchmarks dataset are also in [1]. SODA, finds ontological entities to be aligned using local and semantic similarities. In further research, we plan to tackle alignment of large ontologies.

Function : PHASE3_SEMSIM
Data:
1. O_1 et O_2: two ontologies to align
2. V_{LS}: linguistic similarity vector
3. V_{SS}: structural similarity vector
4. Π_L and Π_S: the respective weights of linguistic and structural similarity

Results: V_{SEMS} : semantic similarity vector
Begin
  /*Parse all nodes of the ontology O_1 */
  forall *($N\text{œ}ud_1 \in O_1$)* **do**
    /*Parse all nodes of the ontology O_2*/
    forall *($N\text{œ}ud_2 \in O_2$)* **do**
      If $Node_1$.`type`==$Node_2$.`type` **then**
        /*Extract the linguistic similarity of $Node_1$ and $Node_2$ from V_{LS}*/
        Sim_L=EXTRACTSIM(V_{LS},$Node_1$,$Node_2$)
        /*Extract the linguistic similarity of $Node_1$ and $Node_2$ from V_{SS}*/
        Sim_S=EXTRACTSIM(V_{SS},$Node_1$,$Node_2$)
        /*Compute the semantic similarity*/
        $Sim_{SEM} = \Pi_L Sim_L + \Pi_S Sim_S$
        /*Add $Node_1$, $Node_2$ and Sim_{SEM} to V_{SEMS}*/
        ADD(($Node_1$,$Node_2$,Sim_{SEM}),V_{SEMS})
  return(V_{SG})
End

Algorithm 3: PHASE3_SEMSIM

References

1. Zghal, S., Ben Yahia, S., Mephu Nguifo, E., Slimani, Y.: SODA : Une approche structurelle pour l'alignement d'ontologies OWL-DL. In: Proceedings of the first French Conference on Ontology (JFO 2007), Sousse, Tunisia (October 2007)
2. Smith, M.K., Welty, C., Mcguinness, D.L.: OWL: Ontology Web Language Guide. Technical report, W3C: Word Wide Web Consortium, http://www.w3.org/TR/2004/REC-owl-guide-20040210/ (February 2004)
3. Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W.R., Yatskevich, M.: Results of the ontology alignment evaluation initiative 2006. In: Proceedings of the First ESWC 2006 international workshop on ontology matching. (2006)

4. Monge, A., Elkan, C.: The field-matching problem: algorithm and applications. In: Proceedings of the 2^{nd} International Conference on Knowledge Discovery and Data Mining. (1996) 267–270
5. Touzani, M.: Alignement des ontologies OWL-Lite. Master's thesis, University of Montreal (2005)
6. Euzenat, J., Loup, D., Touzani, M., Valtchev, P.: Ontology alignement with OLA. In: Proceedings of the 3^{rd} International Workshop : Semantic Web Conference EON, Hirochima, Japan (November 2004) 341–371

Acknowledgements

This work is partially supported by the French-Tunisian project PAI CMCU 05G1412. Thanks to Amine FERJANI and Marouen KACHROUDI for implementing the method.