10

Ontology Alignment OWL-Lite

Aammou Souhaib, Khaldi Mohamed and El Kadiri Kamal Eddine
LIROSA, Faculté des Sciences, Université Abdelmalek Essaadi, Tétouan, Maroc

1. Introduction

The term ontology was originally used in philosophy from the 19i-th century. In this area, it refers to the study of what exists, ie, the body of knowledge about the world (Welty et al., 2001). In the field of knowledge representation, ontologies are considered as relating to different fields of knowledge. They respond to problems of representation and manipulation of knowledge. Ontology is "is an explicit specification of a conceptualization" (Gruber, 1993). Ontologies are widely used in knowledge representation on the Web (Charlet et al., 2004). Nowadays, ontology embodies expert knowledge of a domain. Based on the fact that knowledge can take many different representations, there are nowadays several domain ontologies for the same scope. The alignment techniques represent a general framework in which several ontologies can be exploited.

The alignment also allows the exchange of a semantic point of view, the view of many people (Bach et al., 2004). Although some work on ontologies show the necessity of using domain knowledge (Aleksovski et al., 2006) in certain situations, several methods for ontology alignment that do not have domain knowledge been developed. The main methods are cited: ANCHORPROMPT (Noy et al., 2001), IF-MAP (Kalfoglou et al., 2003), ASCO (Bach et al., 2004), GLUE (Doan et al., 2004), QOM (Ehrig et al., 2004a) and OLA (Euzenat et al., 2004b). These main methods1 exploit ontologies in format markup languages (XML, RDF (S) and OWL-Lite2). In addition, most of these methods exploit similarity measures that cover more or less the whole structure of ontologies to align. These methods generally exploit a threshold of stability provided by the user, to ensure the cessation of the alignment process.

However, this level of stabilization does not spread wide for the calculation of similarity. The OLA is the only method to have the advantage of support for OWL ontologies format-Lie. The OLA method uses a threshold of stabilization to calculate the alignment. Alignment method proposed in this research can implement a new algorithm for automatic alignment of ontologies OWL-Lite. In each pair of entities belonging to the same category, the alignment algorithm calculates the similarity measures. It defines two models for calculating the similarity (local and global), while addressing the problem of circularity and user intervention in the alignment process. The experimental results show an improvement of evaluation metrics from OLA.

The paper is organized as follows. The second section provides a comparative study of the main methods of ontology alignment chosen. In the third section, our method of aligning

ontologies OWL-Lite is described. The fourth section shows an experimental evaluation. The conclusion and future work are the subject of the last section.

2. Comparative study of methods of alignment

Ontologies created can be described in several languages, eg, XML (Marsh, 2001), RDF (S) (Klyne et al., 2004), DAML + OIL (Connolly et al., 2001) and OWL (Smith et al., 2004). The purpose of these languages is to represent the ontologies in a common language. OWL also enables the sharing, import and export ontologies. It is considered the standard ontology for the domain of Semantic Web (Berners-Lee et al., 2001). For these reasons any ontology that is not described in OWL drawbacks. The alignment of two ontologies is to find a match between their entities that are semantically similar (Ehrig et al., 2004b). In a formal way, alignment is defined by the map function as follows:

$$Map : O \rightarrow O' \text{ such that } map(e_1) = e_1' \qquad \text{if } sim(e_1, e_1') > t,$$

Where O and O′ are the two ontologies to align, means a minimum threshold of similarity belonging to the interval [0,1], $e_1 \in O$ and $e_1 \in O'$. This threshold indicates the minimum level for two entities are similar. Each entity e_i is more aligned to a single entity e_j'. Several criteria were used for the comparison of alignment methods, eg, the input format, output format, the measures of similarity and the quality of alignment (Do et al., 2002).

In the remainder of this section we detail the formats for input and output measures of similarity and the quality of alignment.

2.1 Formats in entry and exit

The type of data used must be specified for each method of alignment. Ontologies to be aligned can be represented with languages with beacons or the format of the conceptual graphs. The languages with beacons are XML, RDF(S), DAML+OIL and OWL. The dictionaries of synonymies or lexicons are extra information sometimes being able to be added and which necessary for the improvement of are returned process of alignment.

The format and the structure of the result of alignment are specified for each method. It should be specified if alignment is carried out between the whole structures or couples of entities of two ontologies. The result for the majority of the existing methods is a fi to shit of alignment (generally in format XML), indicating which are the ontological couples entities which correspond. All the methods of alignment determine correspondences between the ontological entities by using measurements of similarity.

2.2 Measurements of similarity

Following taxonomy are proposed for the classification of various measurements of similarity (Rahm and Al, 2001):

i. Terminological method (T): compare the labels of the entities. It is broken up into purely syntactic approaches (TS) and those using a lexicon (TL). The syntactic approach carries out the correspondence through measurements of dissimilarity of the chains (e.g., EditDistance). While, the lexical approach carries out the correspondence through the lexical relations (e.g., synonymy, hyponymy, etc.);

ii. Method of comparison of the internal structures (I): compare the internal structures of the entities (e.g., interval of value, cardinality of attributes, etc.);
iii. Method of comparison of the external structures (S): compare the relations between entities and others. It is broken up of methods of comparison of the entities within their taxonomies (ST) and methods of comparison of the external structures by holding account of cycles (SC);
iv. Method of comparison of the authorities (E): compare the extensions of the entities, i.e., it compares the whole of the other entities which are attached to him (authorities of the classes);
v. Semantic method (M): compare interpretations (or more exactly the models) of the entities.

2.3 Quality of alignment

Measurements of **Precision**, **Recall** and **Fallout** (Do and al, 2002) were the metric ones largely exploited to estimate the quality of alignments obtained. The EON "Evaluation of Ontology-based Tools" (EON, 2004, EON, 2006, Euzenat and al, 2006) retains these measurements for the evaluation of the quality of alignment. The main aim of these measurements is the automation of the process of comparison of the methods of alignment as well as the evaluation of quality of produced alignments. The first phase in the process of evaluation of the quality of alignment consists in solving the problem manually. The result obtained manually is regarded as the alignment of reference. The comparison of the result of the alignment of reference with that of the pairing obtained by the method of alignment produces three units: N_{found}, $N_{expected}$ and $N_{correct}$. The unit N_{found} represents the pairs aligned with the method of alignment. The $N_{expected}$ unit indicates the whole of the couples paired in the alignment of reference. The $N_{correct}$ unit is the intersection of the two units N_{found} and $N_{expected}$. It represents the whole of the pairs belonging at the same time to alignment obtained and the alignment of reference. The precision is the report/ratio of the number of found relevant pairs, i.e., "$N_{correct}$", reported to the full number of pairs, i.e., "N_{found}". It returns thus, the part of the true correspondences among those found. Thus, the function precision is defined by: precision = $|N_{correct}|/|N_{found}|$. The recall is the report/ratio of the number of found relevant pairs, "$N_{correct}$", reported to the full number of relevant pairs, "$N_{expected}$". It specifies thus, the share of the true found correspondences. The function recall is die fi denies by: recall=$|N_{correct}|/|N_{expected}|$. Fallout measurement makes it possible to estimate the percentage of errors obtained during the process of alignment. It is defined by the report/ratio of the erroneous pairs, "$(N_{found} - N_{correct})$", brought back to the full number of the found pairs, "N_{found}", i.e., Fallout = 1-($|N_{correct}|/|N_{found}|$)

Table 1 presents summary and transverse review principal know-discussed methods of alignment. The first entry of table 1 presents the formats of ontologies dealt with by each method of alignment. These formats are as a majority of the languages of beacons except for KIF and OCML. The second entry of table 1 indicates the nature of the fi to shit result which is a fi to shit XML or a fi to shit RDF(S). The third entry of table 1 gathers the various measurements of similarity exploited on the level of each method. The last entry of table 1 puts forward the terminals of measurements of precision for each method within the framework of the tests carried out by EON (EON, 2004). Thus, method OLA compared has a light advantage to method QOM. The "qualitative" performances of these methods are

almost similar, since they take into account all the characteristics of ontology to knowing, the terminological similarity, and structural of entities of ontologies. Moreover, the quality of alignment produces by OLA is better. Indeed, the value of the minimal precision and the value of the maximum precision are higher than those provided by QOM. To note that, OLA proposes a method of calculating of similarity who solves the problem of circularity between the concepts during the process of alignment (Euzenat and Al, 2004b). The result of alignment is appeared as a file RDF/XML.

	GLUE	**OLA**	**IF-MAP**	**ASCO**
Input	XML	OWL-Lite	KIF, OCML, RDF(S)	RDF(S)
Output	XML	RDF	RDF	RDF
Similarity	E	T, TS, I, S, ST, SC, E	ST, E	T, TS, TL, ST
Precision	[0,3-0,6]	[0,6-0,8]	-	-

Table 1. Comparative table of the principal methods of alignment.

In the majority of the principal methods of alignment of ontologies, the stabilization of the measurement of similarity is exploited. This measurement of stability is provided by the user through a threshold. This threshold allows the propagation of the similarity to reach optimal alignment. This propagation is likely not to suitably exploit the vicinity of the various ontological entities. In this way, the method of alignment can stop without exploring of advantage the vicinity. This stop is due to the fact that the treatment of two successive neighbors does not bring a profit lower than the specified threshold. In the same way, the stop limits the treatment of the interesting entities and risk to harm the result of alignment obtained. These disadvantages encouraged us to propose a new method of alignment. The main advantage lies in the fact that it eliminates the intervention from the user by exploiting a wider vicinity of the entities to be paired. The following section introduces the new method of alignment of OWL-Lite ontologies developed which we then compare with method OLA.

3. Our approach to ontology alignment

The method of ontology alignment that we propose takes as input ontologies described in OWL-Lite. OWL-Lite ontologies are transformed to match the form of an OWL-GRAPH that we introduce. The OWL-GRAPH can represent all the information contained in the ontology OWL-Lite (Smith et al., 2004). Classes, properties and instances are nodes in the graph proposed. Nodes in the OWL-GRAPH represent the six types of entities that exist in an ontology OWL-Lite: concepts, instances of concepts, data types, values, data types and properties of classes (such purpose and nature of data type). Relations between entities in the ontology OWL-Lite are the arcs between nodes of the graph. Arcs that exist in the OWL-GRAPH reflect the semantic relationships between entities of an ontology. The OWL-GRAPH is used to represent four categories of specialized links, attribution, instantiation and equivalence. Figure 1 shows an example of two ontologies represented through two separate graphs OWL-Graph. The first ontology indicates that a teacher supervising a student who achieves his memory. The second ontology indicates that a memory is made by a student who is supervised by a teacher. OWL Graphs-Graph obtained by the construction module operated

by the alignment module ontologies OWL-Lite. Indeed, the alignment module performs the course of two ontologies represented in the form of two graphs OWL-Graph. This course compares the nodes and arcs of graphs to determine the correspondences between different ontological entities operating in the diameter of the nodes. The diameter of a node is the number of nodes separating the end of the graph (instances).

The new method of alignment proposed is an approach basing itself on a model of calculation of the similarities local and total. This model follows the structure of the OWL-Graph graph to calculate measurements of similarity between the nodes of two ontologies. The module of alignment associates for each category of nodes a function of aggregation. The function of aggregation takes into account all measurements of similarities between the couples of nodes close to the couple to node to be paired. Thus, this function exploits all the descriptive information of this couple. Table 2 presents the notations used in the developed algorithms. The algorithm which implements the method of alignment proposed takes in entry two ontologies to be aligned in the form of two files OWL-Lite and produces a result in the form of a fi shitting XML.

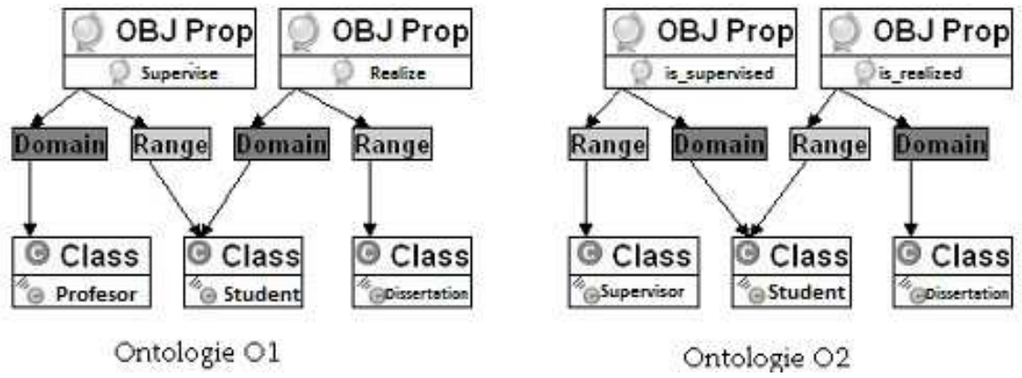


Fig. 1. Example of two OWL-Graph graphs of two ontologies.

- O1, O2: two ontologies to be aligned
- VS_T: terminological vector of similarity
- VS: semantic vector of similarity
- VD: vector of the respective diameters minimum to each couple of nodes

Each node of ontology presents among its characteristics the following fields:

- type: the type of the node
- diameter: the diameter of the node

Each element of the vectors V ST and V S is characterized by the following fields:

- the node of ontology O1
- the node of ontology O2
- the value of similarity

Table 2. Notations used in algorithms PHASE1_SIMTERM and PHASE2_SIMSEM.

It should be noted that our method operates in two successive stages. The first stage, implemented by the means of function PHASE 1_SIMTERM, makes it possible to calculate the local similarity (terminological). The second phase, c.f. function PHASE2_SIMSEM, makes it possible to calculate the similarity total, known as semantic.

3.1 Calculation of local similarity

The calculation of the local similarity is carried out only once for each couple of nodes. The measurement of local similarity of the couples of entities is calculated via algorithm 1 (c.f., function PHASE1_SIMTERM). The calculation of the similarity local (or terminological) is carried out between the descriptors of entities like the names, the comments, etc. the terminological similarity is made up of the syntactic similarity and the lexical similarity. Thus, the syntactic similarity is calculated via the functions of LEVEINSTEIN or EditDistance (Euzenat and Al, 2004a). While the API of WORDNET (Miller, 1995) is exploited for calculation of lexical similarity. Function PHASE1_SIMTERM makes it possible to calculate the terminological similarities of the couples of nodes of two ontologies. It takes in entry two ontologies O1 and O2 to be aligned, represented in the shape of two OWL-Graph graphs, as well as the function of terminological similarity to use and gives in return a vector of terminological similarity of each couple of nodes.

The function CalculSimTerm (Algorithm 1, line 8) takes in entry two nodes N1 and N2, and turns over a value of similarity. This function is provided by one of the methods of calculating of following similarity: the measurement of LEVENSHTEIN, the distance from the under-chains or the API of WORDNET. The local similarity for the various couples of entities is exploited thereafter for the calculation of the total similarity. The following section describes in detail the computing process of the total similarity.

```
 1  Function: PHASE1_SIMTERM
            Data1: Two ontologies O1 and O2
            Data2: terminological Function of similarity
            Results: Vector of local similarity VS_T
 2  Begin
 3          /* course of the nodes of O1 ontology */
 4          For each (N1 ∈ O1) make
 5                  /* course of the nodes of ontology O2 */
 6                  For each (N2 ∈ O2) make
 7                  If N1.type=N2.type then
 8                          Sim_T=calculSimTerm(N1,N2)
 9                          /* add: 2 nodes and the value of the terminological
10                          similarity*/
11                          add ((N1, N2, Sim_T), VS_T)
            return (VS_T)
    End
```

Fig. 2. Algorithme1.PHASE1_SIMTERM.

3.2 Calculation of total similarity

The calculation of the similarity total, known as semantic, is done between the whole of close nodes by categories. Function PHASE2_SIMSEM organizes, by categories, the adjacent

nodes with the couple of entities to be paired. Then, it calculates the measurement of similarity between each of the same pair category. To carry out this calculation, the measurement of similarity "Match-Based similarity" is used:

$$\boldsymbol{MSim}(\boldsymbol{E}, \boldsymbol{E'}) = \frac{\sum_{(\boldsymbol{i},\boldsymbol{i'}) \in \boldsymbol{pair}(\boldsymbol{E},\boldsymbol{E'})} \boldsymbol{Sim}(\boldsymbol{i}, \boldsymbol{i'})}{\boldsymbol{Max}(|\boldsymbol{E}|, |\boldsymbol{E'}|)} \quad (1)$$

where E and E′ represent two whole of the same nodes category. This function, requires that the local similarities of the couples (i,i') are already calculated, gives like result the couples of the unit P = E×E′. The couples (i, i'), intervening in calculation, must present best measurements of similarity. To choose them, there exist two approaches: the algorithm glouton and dynamic programming (Boddy, 1991). The algorithm glouton carries out local choices. Indeed, when he is confronted with a choice, he takes what seems to him the best to advance, and hopes then that the succession of local choices contributes to an optimal solution. While the dynamic programming try to lead to an approach of global optimization. In our algorithm of alignment the algorithm glouton is implemented. Indeed, the algorithm glouton chooses a couple of entities having the greatest similarity and which is higher or equal to the fixed threshold. Then, it removes the two entities of the couple of the table of the similarities. The algorithm continues the checking for each couple until there does not exist anymore couples having a measurement of similarity higher than the threshold.

```
1  Function: PHASE2_SIMTERM
        Data1: Two ontologies O1 and O2
        Data2: terminological vector of similarity VS_T
        Data3: Weight of terminological similarity Π_L
        Results: Vector of global similarity VS
2  Begin
3       /* calculation of the minimal diameter for each couple of nodes */
4       For each (e ∈ VS_T) make
5            VD_i =min (e1_O1.diametre, e2_O2.diametre)
6            /*iterate until reaching the maximum of the diameters belonging
             to VD*/
7            For (it=1; it ≤ Max j∈ [1, V D.size] VD_j; it++) make
8                 /*to traverse the vector of the similarities of the
                  preceding iteration, the vector of similarity of the first
                  iteration is VS_T*/
9                 For (j=0; J < VS.size; j++) make
10                /* verify number iteration and minimum diameter of
                  nodes to be aligned*/
11                If it < VD_j then
12                     Simvois = CalculSimVois (VS_j.N_O1, VS_j.N_O2)
13                     Sim = Π_L × VS_T (j) + Simvois
14                     V Sj = (NO1, NO2, Sim)
15           Return (VS)
16 End
```

Fig. 3. Algorithme2.PHASE2_SIMTERM.

In order to solve the problem of the dependences of similarity, the method of the system of equations at fixed point (Euzenat and al, 2004b) is exploited. It uses a quasi-linear function

which formally allots to each category of nodes a weight Π., being given a category of nodes X and the whole of the relations implied N(X), the measurement of total similarity SimX: X → [0, 1] is defined by:

$$Sim_X(x, x') = \sum_{F \in N(x)} \Pi_F^X Sim_Y\big(F(x), F(x')\big). \quad (2)$$

The function is standardized since $\sum \Pi_F^X = 1$. In our approach of alignment, which we propose, the weights are fixed by defect for each category of nodes. This does not prevent that the user can assign the weights which it wishes. By using the equation (2), to calculate the total similarity of the various categories, a system of linear equations is obtained. The variables of this system are the similarities of the couples of nodes deduced from the equation (1). The resolution of the system of the equation (2) is done by iterations. Iteration 0 of algorithm 2 (c.f. line 10) exploits the terminological similarities, already calculated by intermediary of the algorithme1. Then, iteration 1 of algorithm 2 uses the equation (2) to calculate the total similarities between couples of the same entities categories. Measurements of similarities of the categories intervening in calculation of the similarity of a couple result from the preceding iteration. Thus, the iteration J functions in the same way as the preceding iteration. The calculation of the total similarity of each couple is based to the measures of similarities calculated with the iteration (j-1). In each iteration, the number of candidates to be aligned falls according to the minimum diameter of the couple of node to pair. The exploration of the diameter of each node allows the propagation of the similarity through the vicinity. The principle of this propagation is explained in what follows.

3.3 Propagation of the similarity through the vicinity

Our method carries out a propagation of similarity definitely better than that of OLA. Indeed, in its process of alignment, all the vicinity of the couple of entity to be aligned is integrated in the calculation of similarity. For example, let us consider the figure 1 which presents two ontologies O1 and O2. Being given the couple of entities (Student (O1), Student (O2)), the calculation of the similarity includes the close entities which enter in plays. The calculation of similarity of the couple in question evokes in this example the objectProperty type and varies for two algorithms our algorithm and OLA. Thus, table 3 presents the entities close to the couple (Student(O1), Student(O2)) for, respectively, our algorithm and OLA. Thus, our method integrates measurements of similarity of the couples of entities (supervise (O1), is_supervised(O2)) and (realize(O1), is_realized(O2)) in the calculation of similarity of the couple (Student(O1), Student(O2)), while OLA, is limited to calculate the measurement of similarity between (realizes(O1), is_supervised(O2)). Consequently, the measurement of similarity for this couple is encircled better with our algorithm than with OLA.

Moreover, our method, contrary to OLA, is not based on the stability of the measurement of similarity, by using a threshold die fi nor by the user. Indeed, algorithm OLA carries out successive iterations and in each iteration, measurements of similarities of the entities to be aligned are compared with those of the preceding iteration. If the variation is lower than the threshold, the entities in question are not treated more in the iterations which follow.

neighboring entities	**Student (O_1)**	**Student (O_2)**
For "our method"	Supervise, realize	Is_supervised, is_realized
For OLA	Realize	Is_supervised

Table 3. Table of the entities close to the couple (Student(O1), Student(O2)).

However, this method is likely to make lose couples of entities, whose measurements of similarity can increase the value of the similarity in the iterations later. To cure this problem, our method uses the notion of the diameter, i.e., the depth of the entity in the OWL-Graph graph. Thus, our method does not stop reiterating on a couple of entities only after having exploited all its neighboring structure.

The measurement of similarity of each couple of nodes varies from an iteration to another until it converges. The iteration count in our method is equal at least maxima of the diameters of the candidates to pair. In each iteration, algorithm 2, (c.f., line 12), check candidates to be aligned. The couples of nodes, whose minimum diameter is lower than the number of the current iteration, will not be treated. However, the diameter of each node in the graph must be given. To determine the diameter of a node, two aspects should be considered. The first consists in checking if the graph is directed or not. The second consists in taking account of the circular relations. The algorithm of the calculation of the diameter uses the representation of the OWL-Graph graph of ontology, and makes it possible to determine the diameters of the existing nodes. Moreover, the OWL-Graph graph considered is a graph not directed. However, there exist categories of nodes for which a diameter equal to zero is given. Indeed, these nodes must be only treated in the iteration of algorithm 2 (c.f., line 10), i.e., in the iteration of calculation of the terminological similarity.

These nodes are of nature standard of data (string, non-negative integer, etc.), or value of data (a numerical value, a character string, etc.). The measurement of similarity of each couple of nodes varies from an iteration to another to deal with the information incorporated in the vicinity. The iteration count in our method is equal to the maximum of the minima of the diameters of the candidates to pair.

In the following section, an experimental evaluation of our method is presented.

4. Experimental evaluation of ontology alignment method

Experimental evaluation of the proposed alignment method was conducted on two complementary aspects. The aspect of "intra-method" will focus on evaluating performance, ie, execution time, method vs. the change in the size of the ontologies to align, and the similarity measure used. The second aspect, called "inter-method" to compare the qualitative results obtained by the proposed method vs. the other methods, eg, OLA. As part of experiments conducted some tests provided in the benchmark base available to the community through competition EON (EON, 2004) are used. These tests are described by table 4 (EON, 2004). The ontology base consists of a set of references. It represents a simplified version number of ontological entities compared with real ontologies. Each test case benchmark base highlights a feature of the second ontology to align with the test database. The purpose of the test base is to take care of all aspects that exist in an ontology OWL-Lite and that could have a significant impact on the evaluation metrics of the result of alignment.

4.1 The aspect “intra-method”

In what follows, we will try to measure the evolution of the performances of our method compared to the increase of the composition structural of ontology. Table 5, presents the statistics raised as for three series of tests which were carried out. Indeed, same ontology was used, i.e., the ontology 101 described in table 4. Each test brings an incremental aspect of the composition structural of ontology. The tests carried out are three types of tests. In the

Test	Characteristics of ontology
101	Same basic ontology
103	The not recognized axioms are replaced by their generalization
205	The names of the entities are replaced by their synonyms
222	The hierarchy of the classes is strictly reduced
225	The restrictions of classes expressed by properties were removed
301	Ontology to be compared is real and similar to basic ontology
304	Ontology is also real and similar to basic ontology

Table 4. Ontologies of tests.

TEST1, the ontology of reference is only made up of classes. Thus, it is made up of 33 entities to align. In the TEST2, the 24 properties of nature object are added to the classes.

The number of entities thus becomes 57. In the TEST3, complete ontology is used, i.e., ontology is made up of 97 entities distributed as follows: 33 classes, 24 properties of nature object and 40 properties of standard nature of data. According to the results presented in table 5, the performances of the process of alignment depend on the two following aspects: size of ontologies to be aligned and the choice of the function of terminological similarity. Indeed, the time of execution increases considerably when the number of entities to be aligned increases and conversely. This increase is more considerable on the level of the module of alignment than on the level of the module of construction of the OWL-Graph graph. The choice of the terminological function of similarity also influences over the execution time of the module of alignment. Indeed, the use of a simple function, like that of LEVENSHTEIN, for the calculation of the terminological similarity reduced the time execution. On the other hand, the use of a function more complex as the WORDNET increases considerably the execution time of the process of alignment. This variation is due to the time spent by the algorithm of alignment for obtaining the value of syntactic or lexical similarity. This time this is much more important with the use of WORDNET than with another function of syntactic calculation of similarity like that of LEVENSHTEIN. Indeed, the use of the API WORDNET requires accesses expensive disc to seek synonymies.

	TEST1	TEST2	TEST3
TE: Construction OWL Graph	3,450	4,700	6,780
TE: Our method (LEVENSHTEIN)	110,465	225,677	357,561
TE: Our method (WORDNET)	148,542	301,978	455,843

Table 5. Execution time of OWL-Graph construction and our method in seconds.

4.2 The aspect “inter-methods”

While being based on the quality of alignment (measurement of precision), method OLA had better results (c.f., table 1). In the same way, method OLA exploits ontologies with the OWL-Lite format. For these reasons, method OLA would be used as method of reference in the facet intra-method. Within this framework, it is important to recall that our method carries out a propagation of similarity on all the vicinity of the entities. It exploits the concept of diameter of the ontological entities has fixed to explore the totality of the structure of ontology. The alignment, produced by our algorithm with each test, is compared with the alignment of reference. Thus, the results of measurements of qualities are calculated. Table 6 recapitulates

the results obtained by the two methods of alignment" our method" and OLA (EON, 2004). The best results of the values of precise details of "our method" are obtained when the structures of ontologies are similar or identical, i.e., tests 101, 103, 222 and 225. Thus, "our method" obtains values of precision for these tests which are higher than 0,910. This is explained by the fact why our approach is more effectively exploits the structures of the entities to be aligned. From where, the entities which have almost the same structure are correctly aligned. The results of the tests where the value of precision is less good explains by two aspects. Firstly, our algorithm calculates measurements of similarities of the same entities category. This induces that certain couples of entities are not taken into account by the process of alignment, from where the whole of the pairs belonging at the same time to alignment obtained and the alignment of reference, $N_{Correct}$, is weak. Consequently, the value of precision is weakened. Moreover, the couples which were excluded from the process of alignment can help with the increase measurements of similarities of the couples of close entities and consequently, to increase the number of correctly aligned couples. Secondly, our algorithm does not use in its process of alignment a comparison between the wording or the comments of the entities.

		Precision		Recall		Fallout	
Test	**Similarity**	**"our method"**	**OLA**	**"our method"**	**OLA**	**"our method"**	**OLA**
101	**LEVENSHTEIN**	1,000	0,587	1,000	0,970	0,000	0,400
103	**LEVENSHTEIN**	0,985	0,540	0,985	0,901	0,011	0,430
205	**WORDNET**	0,500	0,470	0,500	0,802	0,493	0,500
222	**WORDNET**	0,917	0,530	0,957	0,901	0,063	0,430
225	**WORDNET**	0,953	0,570	0,953	0,967	0,027	0,400
301	**WORDNET**	0,610	0,483	0,740	0,607	0,347	0,493
304	**WORDNET**	0,592	0,428	0,680	0,618	0,353	0,541

Table 6. Comparison between "our method" and OLA.

In order to evaluate the results of the approach of alignment proposed, table 6 compares the results of two algorithms "our algorithm" and OLA. The statistics obtained are presented in table 6. Starting from the data presented in table 6, our method of alignment is better compared to method OLA. Indeed, our method of alignment provides measurements of more powerful qualities on almost the majority of the tests. These best results are explained by the two following aspects. The first is the fact that "our method" carries out a propagation of similarity definitely better than that of OLA. The second aspect is that our method, contrary to OLA, is not based on the stability of the measurement of similarity by using a threshold ε define by the user. The default value of this threshold is fixed to 0,01 in OLA.

5. Conclusion

In this paper, we presented our method for aligning ontologies OWL-Lite. Alignment method performed to search the best matching pairs by exploiting their respective graphs OWL-Graph. The results obtained by the alignment module are satisfactory compared with results obtained by other methods of alignment. In addition, the proposed method provides better results on most tests compared to the OLA method. A comparison of the execution

time of both methods will be considered to study the scalability of real and complex ontologies. Several improvements are possible on the proposed alignment method to make it more relevant. These improvements include: the calculation richer and fuller of the similarity of terminology, the calculation of inter-category similarity and alignment of ontologies more complex. Finally this method will be integrated into a system of perception of learning activity by a tutor via e-learning platform.

6. References

Aleksovski Z., Klein M., ten Kate W., van Harmelen F., « Matching Unstructured Vocabularies using a Background Ontology », *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management*, Hong Kong, p. 182-197, 2006.

Bach T. L., Dien-Kuntz R., Gandon F., « On Ontology Matching Problems - for Building a Corporate Semantic Web in a Multi-Communities Organization », *Proceedings of ICEIS (4)*, Porto, Portugal, p. 236-243, 2004.

Berners-Lee T., Hendler J., Lassila O., « The semantic Web », *Scientific American*, 2001.

Charlet J., Bachimont B., Troncy R., « Ontologies pour leWeb Sémantique », *Revue I3, numéro Hors Série «Web sémantique»*p. 43-63, 2004.

Connolly D., Harmelen F. V., Horrocks I., McGuinness D. L., Patel-Schneider P. F., Stein L. A., DAML+OIL: Reference Description, Technical report, W3C: Word Wide Web Consortuim, http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218, December, 2001.

Do H., Melnik S., Rahm E., « Comparison of schema matching evaluations », *Proceedings of the 2nd Int. Workshop on Web Databasess*, German Informatics Society, Erfurt, 2002.

Doan A. H., Madhavan J., Domingos P., Halevy A., *Handbook of Ontologies : International Handbooks on Information Systems*, Springer Verlag, Berlin, chapter 18 "Ontology Matching : a Machine Learning Approach", p. 385-404, 2004.

Ehrig M., Staab S., « QOM : Quick Ontology Mapping », *Proceedings of The 3rd ISWC, GI Jahrestagung (1)*, Hiroshima, Japon, p. 356-361, November, 2004a.

Ehrig M., Sure Y., « Ontology Mapping - An Integrated Approach », *in* , C. Bussler, , J. Davis, , D. Fensel, , R. Studer (eds), *Proceedings of the 1st ESWS*, vol. 3053, Springer Verlag, Hersounious, p. 76-91, 2004b.

EONW., « EON : Ontology Alignment Contest », *Proceedings of the 3rd Workshop Evaluation of Ontology-based Tools (EON)*, 2004.

Gruber T., « A translation approach to portable ontology specifications », *Knowledge Acquisition*, vol. 5, n" 2, p. 199-220, 1993.

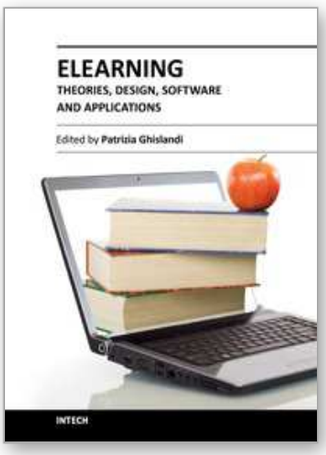
Kalfoglou Y., Schorlemmer M., « IF-Map : an ontology mapping method based on information flow theory », *Journal of data semantics*, vol. 1, n" 1, p. 98-127, 2003.

Klyne G., Carroll J. J., Resource Description Framework (RDF) : Concepts and Abstract Syntax, Technical report, W3C : Word Wide Web Consortuim, http ://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, February, 2004.

Smith M. K., Welty C., Mcguinness D. L., OWL : Ontology Web Language Guide, Technical report, W3C :WordWideWeb Consortium, http ://www.w3.org/TR/2004/REC-owl-guide-20040210/, February, 2004.

Noy N. F., Musen M. A., « Anchor-PROMPT : Using Non-Local Context for Semantic Matching », *Proceedings of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, WA, August, 2001.

Welty C., Guarino N., « Supporting ontological analysis of taxonomic relationships », *Data and Knowledge Engineering*, vol. 1, n" 39, p. 51-74, 2001.



eLearning - Theories, Design, Software and Applications
Edited by Dr. Patrizia Ghislandi

ISBN 978-953-51-0475-9
Hard cover, 248 pages
Publisher InTech
Published online 11, April, 2012
Published in print edition April, 2012

The term was coined when electronics, with the personal computer, was very popular and internet was still at its dawn. It is a very successful term, by now firmly in schools, universities, and SMEs education and training. Just to give an example 3.5 millions of students were engaged in some online courses in higher education institutions in 2006 in the USA1.eLearning today refers to the use of the network technologies to design, deliver, select, manage and broaden learning and the possibilities made available by internet to offer to the users synchronous and asynchronous learning, so that they can access the courses content anytime and wherever there is an internet connection.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Aammou Souhaib, Khaldi Mohamed and El Kadiri Kamal Eddine (2012). Ontology Alignment OWL-Lite, eLearning - Theories, Design, Software and Applications, Dr. Patrizia Ghislandi (Ed.), ISBN: 978-953-51-0475-9, InTech, Available from: http://www.intechopen.com/books/elearning-theories-design-software-and-applications/ontology-alignment-owl-lite

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen