23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Ontology Alignment using Stable Matching

Imene Ouali[a*], Faiza Ghozzi[a], Raouia Taktak[b], Mohamed Saifeddine Hadj Sassi[b]

[a]*MIRACL Laboratory, ISIMS, University of Sfax, B.P. 242, 3021 Sakiet Ezzit, Sfax, Tunisia*
[b]*Digital Research Center of Sfax, LT2S Laboratory, TUNISIA*

Abstract

Ontology matching is a solution to the problem of semantic heterogeneity in the integration and sharing of information. It consists in establishing mappings between entities which semantically belong to different ontologies. Most ontology mapping approaches use elementary matching techniques (eg, string-based methods, linguistic methods, etc.). These techniques map the elements by analyzing the entities in isolation and ignoring their relationships with other entities (father/son, brother, etc.). Bypassing the latter aspects, the determination of the semantics of an entity is often difficult. Hence, the structural information of an ontology plays an important role in ontologies mapping. In this work, we study methods of structural alignment. We adopt two alignment methods that are based on the structure. The first, called Method of Similarity of Inheritance (MSI), applies the initial similarity method based on the concepts and integrates the inheritance relation (father/son) into the calculation of similarity. The second, called Method of Sibling Similarity (MSS), involves sibling relationships to enrich the similarity score between two concepts. Moreover, we enhance the mappings selection strategy by integrating the stable marriage algorithm as an optimal matching strategy. This algorithm improves the ontology alignment quality through efficient optimization techniques and creates a faster and more robust alignment method.

Keywords: Matching, ontologies, alignment, mapping, semantics, structural, stable marriage.

1. Introduction

With the advent of the web and Internet, the number of users has increased exponentially, generating a large volume of data. These data are part of several application domains such as e-commerce, finance, medicine, biology, data warehouses, etc. This multidisciplinarity has lead to the appearance of a heterogeneous set of data, thus interrupting the process of information exchange between the application domains. In order to deal with this data heterogeneity, one may use the integration, mediation and standardization of these data. A fairly effective way has been put in place

E-mail address: Imene3ouali@gmail.com

to help standardize: the ontologies. Ontologies can be used to describe the structure and semantics of the data. They aim at overcoming problems of semantic ambiguity in the different domains, making them interoperable. In particular, ontology alignment is a process for finding semantic relations between classes of a source ontology and classes of a target ontology. It allows to find similarities between the entities of semantically related ontologies in order to determine mappings between their concepts.

In this paper, we devise an enhanced matching approach that allows to obtain a high level of alignment quality from two ontologies. We propose two matching methods based on the graph structure. We not only consider labels but also relationships between the concepts and their positions in the ontology tree. The first method, called Method of Similarity of Inheritance (MSI), emphasizes father/son relationships when calculating the similarity between two concepts. The second method, called Method of Sibling Similarity (MSS), takes into consideration the sibling relations of the concepts. In addition, we propose an improvement of the mappings selection strategy by including efficient optimization techniques through the stable marriage algorithm used to enhance the ontology alignment quality.

The paper is organized as follows. In Section 2, we define the context of our work by exposing some related works. In Section 3, we highlight the main research issues of ontologies matching. Then, we present our matching approaches in Section 4. A discussion of the obtained results as well as a comparison with some previous research works are exposed in Section 5. Finally, concluding remarks are presented.

2. Related Work

In this section, we present some related works with a focus on the development of new methods and systems of correspondence. A comparative study of different methods and systems is presented through the section and a summary of similarity methods is given in Table 1. These methods and systems will be used as a reference for hypotheses and algorithms in our contribution.

Table 1: Comparison between similarity methods

Method	Lexicon	Structure	Hierarchy	Semantics	Heuristic
[1]	X	X			
[2]	X	X	X		
[3]	X	X			
[4]		X		X	X
[5]	X	X			
[6]	X			X	X
[7]	X			X	
[8]	X	X	X	X	
[9]				X	X
[10]	X	X		X	

We present several approaches related to structural, semantic and hybrid similarity techniques. Several ontology matching methods have been proposed and devised in order to obtain stable alignments of ontologies. These methods are either based on a lexical analysis (terminology), adopt a structural approach such as in [3, 4] or hybrid (combining structural and semantic) such as in [10, 8]. In our study, we limit ourselves to these three categories of approaches. The extensive approaches are not considered. Recall that the lexical method considers concept labels to align concepts. The structural and semantic methods support lexical approaches in order to find mappings in input ontologies.

Nguyen and Conrad [1] propose a method that is composed of two measures. The first measure is the basic lexical similarity measure, i.e., edit distance metric, that provides initial mappings. The second one is a structure-based similarity measure that enables to find mappings between ontology concepts. In [2], Mork and Bernstein propose an approach to align two ontologies of human anatomy. This approach is composed of three phases: lexical alignment that identifies concepts with similar names, structural alignment which present concepts whose neighbors are similar and hierarchical alignment that identifies concepts with similar descendants. In [3], Xie et al. propose a method that

is composed of four processes named ontology as a graph, matching terminology, structural matching and candidate selection. In [4], the authors propose a measure of similarity based on the structure for ontology matching using a threefold method. The first step enables to establish preliminary mappings between any two entities from two given ontologies. The second step is to select the centroid matches of the two ontologies. The third step is a similarity propagation method based on the structure. In [5], authors define a structural similarity measure that combines the similarities for three (1: 1) matcher. These matches are ISub, VDoc and GMO. In [6] authors propose an extension of the alignment system called Ontologies Falcon-AO. Furthermore, in [7], authors define an Ontology Mapping For Information Management (OMFIM) algorithm that is evaluated with OAEI (Ontology alignment Evaluation Initiative) benchmark dataset. The performance of the algorithm is compared against S-match algorithm in order to solve the problem of semantic heterogeneity. In another work [8], the authors define a method for the combination of three different types of similarity measures: lexical techniques, structure-based techniques, and semantic-based techniques. In addition, authors of [9] define a measure that focuses on matching ontologies based on the structure and aggregation of similarities calculated by different matches. In [10], authors present a measure of similarity based on the structure for ontology matching by the terminological, structural and semantic level.

3. Ontology matching issues

Recently, various methods have exhibited similar properties for different systems. In spite of the diversity of methods, current ontology alignments does not give (for some test cases) effective results from a quality and performance point of view (see for example [11]).
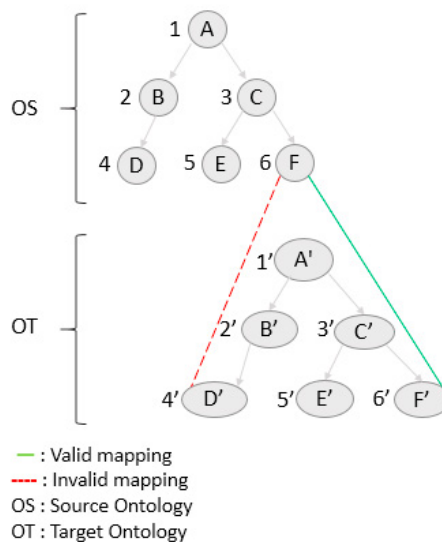

Fig. 1: Mapping example

In this section, we propose a new structural alignment solution integrating neighborhood relationships (father/son, brothers) and an approximate mapping selection algorithm based on the stable marriage approach. In fact, the structural methods translate the principle that two entities within a structured representation are more similar than the entities that surround them [12]. Thus, the similarity of each pair of entities is dependent on the similarity values of its neighbors and subsequently induces an approximate recursive calculation similar to the transmission of similarities between neighboring pairs. As a consequence, the basic similarity between the concepts of the source ontology and the concepts of the target ontology will not be sufficient to obtain a high degree of precision. To solve this problem, we must take into consideration the structure of the ontology to find correct mappings. Therefore, we consider the inheritance relationships between the descendants (fathers and sons) and sibling relationships (brothers). Figure 1 shows an example of mapping between concepts with the same name. In this example, concept 6 of the source ontology (OS) can be mapped to concepts 4′ and 6′ of the target ontology (OT). Considering that the fathers of concepts 6 and 6′

have the same name we can select $(6, 6')$ as the valid alignment. In fact, alignment $(6, 4')$ is an incorrect alignment inducing a poor accuracy.

The second problem is the choice of the mappings selection method. This choice influences the results. Since our alignment method is of type (1: 1), choosing the max of similarities of the candidate alignments between a source concept and all the concepts of the target ontology is the most applied solution. Merging a concept s (from the source ontology) to a concept t (from the target ontology), based on the fact that $sim(s, t)$ corresponds to the max of the similarities between s and the set of all concepts of target ontology, can lead to unstable coupling. The non-stability implies that s can be coupled with c of the target ontology, while t has another "preferred" concept such as $sim(s', t) > sim(s, t)$. The *stable marriage* algorithm tries to solve this problem by proposing an approximate solution that improve coupling stability between the concepts of the two ontologies. This is defined in the overall goal of improving the quality of alignment.

4. Architecture of the proposed matching methods

This section gives a detailed description of the three phases of our approach which are the pre-matching, the matching and the post-matching phases.

4.1. Pre-matching phase: Initial Similarity

This phase presents is a pre-processing of ontologies. It consists in standardizing ontology concepts in order to determine whether they are similar or not. In this phase, we establish initial mappings between the concepts of the source ontology (OS) and the concepts of the target ontology (OT) by using a similarity algorithm named SimI (Initial Similarity). If the returned value is greater than or equal to a threshold (S), that is chosen within the user requirements, then both concepts would be matched. For each concept s of (OS), we note by Mapp(s) the set of concepts t of (OT) that correspond to s such that $SimI(s, t) \geq S$. Mapp(s) is called *mapping* of s.
The following steps determine the similarity between two concepts s and t. We may refer to these concepts by their labels (s) and (t), respectively.

- If the label (s) is identical to the label (t) then *SimI*(s, t) will be equal to 1.

- Otherwise, we apply the processing function (*i.e.*, F_process_chain) on the label (s) and the label (t).

The processing function F_process_chain is used to separate a compound string that contains several words into its components. For example, in an ontology that contains related concepts, if tag (s) = "bouquet_of_flowers", the string of the tag will be converted to "bouquet of flowers" after the function is applied. Similarly, if the tag (s) = "saladwithfruit" in an ontology containing related concepts, it will be converted to "salad with fruit" after the application of the processing function. After applying the processing function on the string of label (s) and label (t), we check if the two resulting labels are identical.

In the next step, we remove all the "stop words" (such as "the", "to", "and" ...) from labels (s) and (t) using the function F_delete_stop_word. Besides, we look for the definitions of each word remaining in the tags from the WordNet dictionary (https://wordnet.princeton.edu/). Finally, we apply the derivation algorithm (*stemming*), resulting in sp_source and cp_target that are derived from (s) and (t), respectively. This algorithm reduces words at their root. For example, it reduces the words "drinks" and "drinking" to "drink", we use the function F_ root to this end.

The initial similarity between the two concepts s and t is then calculated as follows [13]:

$$SimI(s, t) = \frac{2 \times |common(sp\_source, cp\_target)|}{long(sp\_source) + long(cp\_target)}, \tag{1}$$

where $|common(sp\_source, cp\_target)|$ is the number of common words between the source concept strings and the target concept strings. $long(sp\_source)$ and $long(cp\_target)$ denote the number of words in sp_source and cp_target, respectively.

4.2. Matching phase

After processing the pre-matching phase, we obtain a set of elements that are linguistically similar. We will use in this phase the two-to-two matching process which consists in determining the mappings between the ontology pairs. For this, we use structural matches to determine the structural similarities between the elements as detailed in following sections.

4.2.1. Inheritance similarity method (MSI)

In the Inheritance Similarity Method (MSI), we focus on the structure-based technique to improve the alignment results that were obtained from the concept-based initial similarity (SimI) method. Therefore, we do not only consider the label of the concepts, but also the positions of the concepts in the hierarchy. This main characteristic plays an important role in the identification of the concept. This means that the (MSI) method is a structure-based technique which uses ancestor relationships.

We now present the details of determination of the $MSI\_sim(s, t)$.

Let us come back to the example of Figure 1. We calculate Parent_i as follows: parent_1(C)=B; parent_1(C')=B'; parent_2(C)=A and parent_2(C')=A'. We also define p as the percentage of contribution which is a parameter that we set at 0.75. This percentage is the fraction of the similarity measure between s and t that will be used to determine the $MSI\_sim(s, t)$. The $MSI\_sim(s, t)$ can be defined as follows [13]:

$$MSI\_sim(s, t) = p \times SimI(s, t) + \frac{2 \times (1 - p)}{n \times (n - 1)} \sum (n + 1 - i) \times SimI(Parent\_i(s), Parent\_i(t)), \quad (2)$$

where $n = \min(depth(s), depth(t))$, and $depth(s)$ ($depth(t)$) is the depth of concept s (t) in the source (target) ontology tree. The depth is nothing but the number of arcs between the concept and the root of the ontology tree. Parent_i (s) (resp. Parent_i (t)) is the ith concept encountered while reaching s (resp. t) from the root of the source ontology (OS) (resp. the target ontology (OT)).

The MSI_sim method gives importance to parents and in general to any ancestor of a concept while calculating similarity between the concepts. The father of a concept is more important than its grandfather. This is achieved by giving a relatively high percentage value for p. The grandfather of concept is more important than its great grandfather, and so on, until the root is reached. We observe the need for the application of the MSI_sim method for the example of Figure 1. In fact, we can notice that the first mapping (*i.e.*, $(6, 4')$) is incorrect since it does not have the same father but the second (*i.e.*, $(6, 6')$) is correct because both concepts have the same father. Thus, we conclude that the MSI_sim method is useful for determining more precise similarity measures between concepts. For the example of Figure 1, the calculation of the MSI_sim (C, C') requires to calculate beforehand the initial similarity SimI (C, C') and is given by $MSI\_sim(C, C') = 0.75 \times SimI(C, C') + 0.17 \times SimI(B, B') + 0.08 \times SimI(A, A')$.

4.2.2. Siblings similarity method (MSS)

In this method, the brothers of a concept contribute to its identification. This can improve the quality of the alignment process. As the MSI_sim method, the MSS_sim method recognizes the similarities between the concepts of the same level. We denote similarity MSS_sim between a concept s of (OS) and a concept t of (OT) by MSS_sim (s, t). In order to give the formula of this similarity, we need first to define some notations. Let the function getNeighbors() give the neighbors (concepts of the same level). For the example given in Figure 1, getNeighbors(C)={D, F}; and get-Neighbours(C')={D', E', F'}. If getNeighbours(s) and getNeighbours(t) are not empty, MSS_sim (s, t) can be defined as follows [13]:

$$MSS\_sim(s, t) = p \times SimI(s, t) + \frac{1 - p}{n} \sum \max\{SimI(s_i, t_1)...SimI(s_i, t_m)\}, \quad (3)$$

where $n = |getNeighbours(s)|$ and $m = |getNeighbours(t)|$.

The main feature of the MSS_sim method is to allow siblings of a given concept to play a role in the concept identification process. The MSS_sim similarity is computed between the concept s in the source ontology (OS) and the concept t in the target ontology (OT) when applying the S_sim method with a p value equal to 75%. Let us go back to the example given in Figure 1 and look for the MSS_sim between C and C'. This similarity is computed by summing 75% of the base similarity between C and C', 12.5% of the maximum base resemblance between {D and D'},{D and E} and {D and F'}, and 12.5% of the maximum base resemblance between {E and D'}, {E and E'} and {E and F'}. The value of p fixed to 75% was chosen following a list of experiments carried as in [13]. Note also that in order to compute MSS_sim (s, t), we need first to find SimI (s, t).

$$MSS\_sim(C, C') = 0.75 \times SimI(C, C') + ((0.25/2) \times (max(SimI(D, D'), SimI(D, E'), SimI(D, F'))) + ((0.25/2) \times max(SimI(E, D'), SimI(E, E'), SimI(E, F'))) .$$

4.3. Post-matching phase: stable marriage

During the phase of post matching, we choose to adapt a method of the so-called stable matching inspired from the stable marriage problem, in order to converge to the most stable situation obtained from the pre-matching and the matching phases. The stable marriage problem can be stated as follows. Given two finite equally sized sets M and W such that each element $m \in M$ has an ordering of preferences of elements of W, and vice versa, the problem consists in finding a stable matching between elements of M and W. M and W refers to Men and Women, respectively. A set of male-female pairs $C \subset M \times W$ is a pairing. If each element of M and W appears in at most one element (pair) of C, in this case the matching C is said to be *perfect*. Each man $m \in M$ and woman $w \in W$ is associated with a total order on the opposite sex set. This order represents his preferences. When m prefers w to w', we write $w >_m w'$. Similarly, when w prefers m to m', we write $m >_w m'$. The perfect matching C is said to be stable when:

$$\forall (m, w), (m', w') \in M \times W, \quad \neg (w' >_m w \wedge m >_{w'} m'). \tag{4}$$

This relation ensures that each man m does not prefer the wife w' of another man m', while this latter prefers the first's wife w as well. This unstable situation leads in fact to separate couple.

The stable marriage problem can be solved using *Gale and Shapley* algorithm (1962) [14]. During the processing of the algorithm, couples are formed. These couples can still be defeated during the execution. The *Gale and Shapley* algorithm is asymmetrical with respect to both sexes: it gives the initiative to one of the two parts. The analysis of this algorithm makes possible to prove the existence of a stable matching. If we have n men and n women then the number of iterations of the main loop is limited by n^2. With well-chosen preferences, it is possible to show that it sometimes takes $O(n^2)$ iterations to finish. So the complexity of this algorithm is in the worst case equal to $O(n^2)$.

In our work, we adapt the stable marriage algorithm to our ontology matching problem. The two ontologies, source and target, are considered as the sets M and W, and the different concepts are the elements m and w to be matched. This phase's objective is to determine the final mappings between ontology elements. It allows to select stable mappings. Therefore, we propose an adaptation of the stable marriage algorithm as described in details in Algorithm 1.

To execute the Stable Marriage algorithm [15] which is known as the deferred acceptance algorithm, we define some variables as follows. Let s and t be strings representing concepts from the source ontology (OS) and the target ontology (OT). s and t are considered as a man and a woman, respectively.

In Algorithm 1, we match men with women according to the preference list of each of them, hence matching concepts with their preferred (or most similar).

Algorithm 1 stable marriage

---
h!]
Variables
s1, s2, t1, t2: character string
source_free, get_Next_preferred, Current: list of character strings
Start
 1: **while** (*source_free* ≠ "") **do**
 2:     Remove s1 from source_free
 3:     t1 takes get_Next_preferred s1   // t1 is the first favorite target by s1
 4: **end while**
 5: **if** (current [t1] = "") **then**
 6:     Current [t1] takes s1   // s1 and t1 are mapped
 7: **else**
 8:     Current [t1] takes s2
 9:     **if** (t1 prefers s1 to s2) **then**
10:         Current [t1] takes s1   // if it prefers s1 to s2, s1 and t1 are mapped
11:         Add s2 to source_free   // s2 becomes free, it is added to the list
12:     **else**
13:         s1 to source_free   // s1 added to the list of free sources
14:     **end if**
15: **end if**
16: **return** Current

In what follows, we present an example for stable marriage applied to ontologies matching. Consider two pairs, the first pair is {*s*1, *s*2} and the second one is {*t*1, *t*2}, where:

- The preference list of s1 is [t1 then t2];

- The preference list of s2 is [t2 then t1];

- The preference list of t1 is [s1 then s2];

- The preference list of t2 is [s2 then s1].

In this example, stability imposes that s1 is coupled to t1 and that s2 is coupled to t2 in order to avoid unstable mapping. s1 and s2 are two source concepts of the source ontology (OS). t1 and t2 are two target concepts of the target ontology (OT). Each concept has a list of preferences that is computed using similarity between concepts. We denote "source _free" the list of source concepts that are not currently mapped. We also denote by "current" the list of sources mapped to targets. If the source concept is free, it will fill a preference list of the preferred target concepts (t1 then t2, etc.) from the most preferred to the least preferred.

Assume s1 requests a mapping to t1, t1 will be mapped to s1 if it is free. Otherwise, we will check the preference of t1, and test the stability s1 with respect to s2. If t1 prefers s2 (with which it is already mapped) to s1, then s1 remains always free. In addition, s2 and t1 will keep their mapping without any modification. However, if t1 prefers s1 to s2, s1 will be mapped to t1 and s2 will become free again.

5. Computational results and discussion

5.1. Evaluation

In this section, we evaluate the contribution of our alignment approach. First, we study the impact of the structural alignment algorithm integrating siblings and inheritance relationships. Then, we focus on the contribution of the stable marriage algorithm during the selection phase. Computations were conducted to study the performance of the used

approaches compared to existing ones. In particular, we compare efficiency in terms of F-measure. Experiments show that our methods give reasonably efficient results.

In order to evaluate the performance of our approach, we based our computations on ontologies taken from the OAEI *Conference* (`http://oaei.ontologymatching.org/2017/conference`). We also tested our approach on some random ontologies that we created to make some small tests. For most of the tests, we focus on seven interesting ontologies as shown in Table 2.

Table 2: Ontologies of tests

Number	Name	Number of classes
O1	Edas	104
O2	Iasted	140
O3	Ekaw	74
O4	sigkdd	49
O5	ConfOf	38
O6	Cmt	36
O7	Conference	59

We apply the same approach for several pairs of ontologies (from p1 to p21) as shown in Table 3 in order to evaluate the effectiveness of our alignment methods. We test different configurations of our approach and compare the results of these different configurations by taking into consideration the F-measure. Note that configurations compute one-to-one mappings between ontologies.

Table 3: Pairs of test ontologies

Name	Source Ontology	Target ontology
P1	Cmt	Conference
P2	Cmt	ConfOf
P3	Cmt	Edas
P4	Cmt	Ekaw
P5	Cmt	Iasted
P6	Cmt	Sigkdd
P7	Conference	ConfOf
P8	Conference	Edas
P9	Conference	Ekaw
P10	Conference	Iasted
P11	Conference	Sigkdd
P12	ConfOf	Edas
P13	ConfOf	Ekaw
P14	ConfOf	Iasted
P15	ConfOf	Sigkdd
P16	Edas	Ekaw
P17	Edas	Iasted
P18	Edas	Sigkdd
P19	Ekaw	Iasted
P20	Ekaw	Sigkdd
P21	Iasted	Sigkdd

Figure 2 illustrates the comparison of the F-measure results obtained for the SimI basic method, the combined structural approach (MSI and MSS), the stable marriage approach and, finally, our global approach integrating the

combined structural approach and the stable marriage. Results are represented while starting scale with 40%. First, we evaluate the contribution of the structural approach by integrating the sibling relationship into the matching. Second, we integrate the inheritance relationship. Then, we add both of inheritance and sibling approaches. Furthermore, we analyze the stable marriage contribution on the same examples. Finally, we test the contribution of both structural approaches and stable marriage.

- ■ Without the integration of siblings and inheritance
- ■ With the integration of siblings and inheritance
- ■ Without the integration of the stable marriage approach
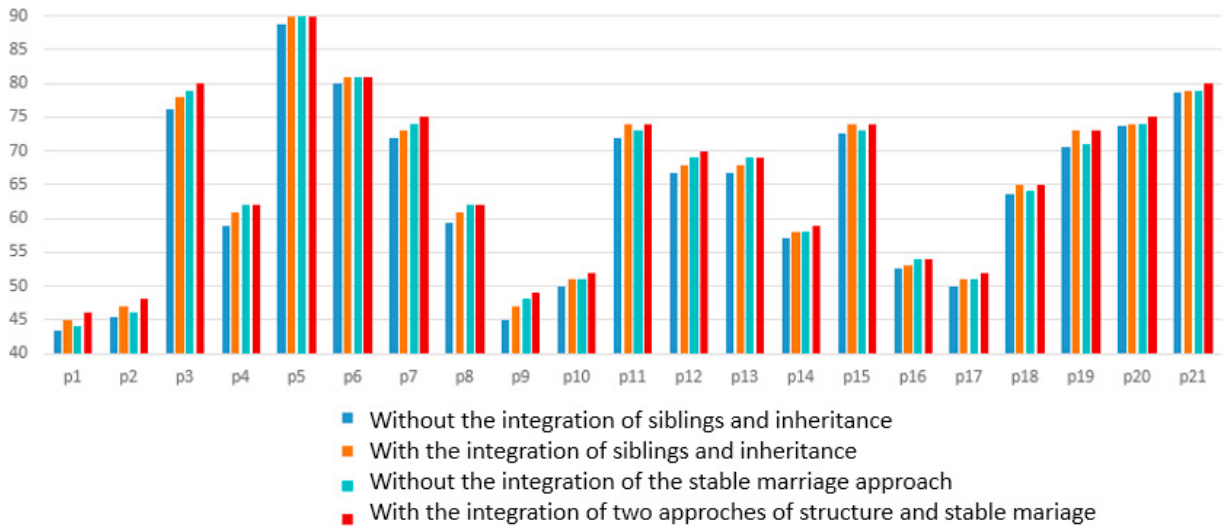- ■ With the integration of two approches of structure and stable mariage

Fig. 2: Integration of the two structural approaches and stable marriage.

Through Figure 2, results show that out of seven ontologies (which make up 21 pairs of ontologies), the configuration that uses the structural and/or stable marriage approaches is often more efficient than the configuration that does not use them. Therefore, the use of these methods improve performance and show that the matched ontologies are not structurally very different.

5.2. Comparison of our approach with other algorithms

In order to evaluate our approach (the global method that includes both structural and stable marriage approaches), we compare our results to the published results of the 2017 OEAI Algorithmic Alignment Evaluation Initiative (`http://oaei.ontologymatching.org/2017/conference/eval.html`). Our own algorithm is compared with different algorithms that are AML, LogMap [16], XMap [17], StringEquiv, and PoMap. Recall that the algorithm that performed best in the 2017 competition was AML. Our algorithm ranks fourth in the ranking of the best algorithms after XMap algorithm. Table 4 illustrates the different scores of all the tested methods.

Table 4: Comparison between our approach and other ontology matching systems

Methods	Precision	Recall	F-measure
AML	0.83	0.7	0.76
LogMap	0.84	0.64	0.73
XMap	0.84	0.64	0.73
Our approach	*0.84*	*0.54*	*0.66*
StringEquiv	0.88	0.5	0.64
PoMap	0.88	0.47	0.61

In this study, we compare the performances of our approach to others from the literature. We can conclude that our algorithm gives efficient results and is significantly better than some other algorithms.

6. Concluding remarks

Ontology Matching establishes mappings between the entities which are semantically different. In practice, many matching methods have been implemented since several years. Most ontology mapping approaches use element matching techniques. They map the elements by analyzing the entities in isolation, and ignoring their relationships with other entities (father/son, brother, etc.). Since the determination of the semantics of an entity is often difficult, the structural information of an ontology is used to improve ontologies mapping.

In this work, we studied structural alignment methods. We have adapted two alignment methods that are determined on the structure. The first method MSI (Inheritance Similarity Method) uses initial similarity based on concepts, and includes inheritance relationship (father/son) in the similarity calculation. The second method MSS (Siblings Similarity Method) enrich the similarity score between two concepts by including the siblings relationship. Furthermore, we use the stable marriage method as an optimal matching strategy in order to obtain stable mappings.

Many extensions can be considered for the future. We are currently conducting experiments to test the effectiveness of our structural alignment method coupled with stable marriage on other large scale ontologies such as HOBBIT Link Discovery (hobbit), Large Biomedical Ontologies (broadbio), Disease and Phenotype (phenotype) etc. Our contribution will be applied to real, complex and highly heterogeneous ontologies. It would also be interesting to create faster and more robust alignment methods. Another interesting approach that can be proposed is to use semantic resources like WordNet to improve the alignment result and thus ensure better efficiency.

References

[1] Thi Thuy Anh Nguyen and Stefan Conrad. A new structure-based similarity measure for automatic ontology matching. In *KDIR*, pages 443–449, 2012.

[2] Peter Mork and Philip A Bernstein. Adapting a generic match algorithm to align ontologies of human anatomy. In *Data Engineering, 2004. Proceedings. 20th International Conference on*, pages 787–790, 2004.

[3] Cheng Xie, Melisachew Wudage Chekol, Blerina Spahiu, and Hongming Cai. Leveraging structural information in ontology matching. In *30th International Conference on Advanced Information Networking and Applications (AINA), 2016 IEEE*, pages 1108–1115, 2016.

[4] Ying Wang, Weiru Liu, and David A Bell. A structure-based similarity spreading approach for ontology matching. In *International Conference on Scalable Uncertainty Management*, pages 361–374, 2010.

[5] Mohammad Mehdi Keikha, Mohammad Ali Nematbakhsh, and Behrouz Tork Ladani. Structural weights in ontology matching. *arXiv preprint arXiv*, pages 41–58, 2013.

[6] Baththama Bello Alhassan, Sahalu B Junaidu, and A Obiniyi. Extending an ontology alignment system with a lexical database. *Scientific Research Journal*, pages 12–17, 2015.

[7] Kaladevi Ramar and TT Mirnalinee. heterogeneous information management using ontology mapping. pages 2078–2081, 2015.

[8] Thi Thuy Anh Nguyen and Stefan Conrad. Ontology matching using multiple similarity measures. In *Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), 2015 7th International Joint Conference on*, pages 603–611, 2015.

[9] Floriana Esposito, Nicola Fanizzi, and Claudia d'Amato. Recovering uncertain mappings through structural validation and aggregation with the moto system. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1428–1432, 2010.

[10] Aroua Essayeh and Mourad Abed. Towards ontology matching based system through terminological, structural and semantic level. *Procedia Computer Science*, pages 403–412, 2015.

[11] Jean-François Djoufak-Kengue, Jérôme Euzenat, and Petko Valtchev. Alignement d'ontologies dirigé par la structure. In *Actes 14e journées nationales sur langages et modèles à objets (LMO)*, pages 43–57, 2008.

[12] Gilles Bisson. Learning in fol with a similarity measure. In *Proceedings of the National Conference on Artificial Intelligence*, pages 82–82, 1992.

[13] Isabel F Cruz and William Sunna. Structural alignment methods with applications to geospatial ontologies. *journal : Transactions in GIS*, pages 683–711, 2008.

[14] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, pages 9–15, 1962.

[15] Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*. 2013.

[16] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *International Semantic Web Conference*, pages 273–288, 2011.

[17] Warith Eddine Djeddi and Mohammed Tarek Khadir. Xmap: a novel structural approach for alignment of owl-full ontologies. In *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, pages 368–373, 2010.