An Approach for the Emerging Ontology Alignment based on the Bees Colonies

Carlos Rangel[1], **José Aguilar**[1,2], **Mariela Cerrada**[1,3], **Junior Altamiranda**[1]
[1]CEMISID, Universidad de Los Andes, Mérida, Mérida, Venezuela
[2]Prometeo Researcher, Universidad Técnica Particular de Loja, Loja, Loja, Ecuador
[3]Prometeo Researcher, Universidad Politécnica Salesiana de Cuenca, Ecuador

Abstract - *This work is a research in the field of Ontologies Integration from the point of view of Ontology Mining based on Services. Specifically, the work focuses on an automatic suggestion of ontological alignments for users. The Ontology Mining area (OM) is very recent, due to the current trend of using ontologies as a mechanism for representing knowledge, which has created a wide field to explore and extract knowledge. The problem lies in the comparison of existing ontologies, in order to use them together, that is, finding their semantic equivalences. There are different techniques for ontologies alignment as a form of ontological comparison based on the matching of the concepts, which is a fundamental process in the ontologies integration. Each alignment technique uses different strategies; based on specific principles, which make it more adequate for a particular context. This paper proposes an automatic approach for comparison and selection of alignment techniques, given a group of ontologies, based on the ABC algorithm, which is inspire by bee colonies. The approach uses as comparison and selection criteria the execution time, the number of aligned concepts, and the number of times the colony chooses each technique (this is due to the stochastic approach of the ABC algorithm).*

Keywords: Ontologies Alignment, Alignment Techniques, Ontologies Integration, Artificial Bee Colonies, Ontology Mining.

1 Introduction

Data Mining (DM) allows the knowledge extraction from historical data. The new trend, giving semantic content to data, gives new branches in DM. Particularly, the Semantic Data Mining (SDM), which allows using DM techniques to extract knowledge from databases, in order to explore the semantic content of the web (called Semantic Web Mining (SWM)). Finally, explore or extract knowledge in the form of ontologies (called Ontology Mining (OM)).

Due to the big development of the ontologies, it is emerged the need for integrating them. In the literature, as a first step to compare the ontologies that will be integrated, it has been develop several ontologies alignment techniques. Some of these techniques are classes' structure, distance of names, and the name properties analysis, among others.

The problem is to determine which alignment technique should be used at a given context, which alignment technique is the best for the ontologies to be integrated. This paper proposes an ABC based technique, which automatically select the proper alignment technique based on the characteristics of the ontologies that are going to be integrated. This algorithm is a first step towards the automation of the process of integration of ontologies.

This article has five sections, the first one summarizes some aspects about Ontology Mining and Semantic Mining, and the second one presents the bases of the ABC algorithm. In the third, it is presented our ABC approach for the Emergent Selection of the Alignment technique, and finally, some conclusions are presented.

2 Ontology Mining (OM)

OM is known as the set of DM techniques for extracting behavior patterns, knowledge, among other features, in order to build or enrich ontologies. Currently, the growth in the amount of available ontologies on a given knowledge domain, has demanded to OM to explore techniques that can extract additional knowledge of a set of ontologies, particularly the integration patterns of several ontologies, in order to create a broader knowledge domain.

We particularly are interested in two integration mechanisms:

- Ontology linking: it is the process of finding relationships between entities belonging to different ontologies [4]. The results can be used to display maps, transforming a source into another (an ontology into another ontology), creating a set of relationships or rules between ontologies, or generating queries for the two ontologies to extract information.

- Ontology Merge: it is the process where multiple ontologies in the same domain must be joined (fused or merged) in order to standardize knowledge, to have full locally knowledge, among others. Ontology Merge is very important in the distributed systems, because it is costly carry out ontologies queries at different sites. These techniques of ontologies mixture is for the case where the ontologies handle the same knowledge, but with different representations, or having partial representations of such knowledge, such that ontologies may share certain concepts and others not. That requires the presence of an expert that has to be present at the time of the mixture, for decision-making.

These ontological integration mechanisms require a process of ontological comparison, which is usually called alignment [11]. The ontology alignment process usually needs the following elements: two ontologies O1 and O2, a set of parameters p, a set of resources r for alignment, and a function of alignment f, which returns a set of correlation A [6].

The function f defines the process that compares different ontological resources (concepts, relations, among others), in order to find correspondences between two concepts. In each O1 and O2 are analyzed each one of its elements: concepts, concepts properties, concept hierarchy, among others. The set p represents the requirements for alignment, p = {design language like OWL (Web Ontology Language), number of elements, vocabulary, among others}. The set of resources refers to the elements that are used to obtain the set of correlation (r = {similarity measures, algorithms}). The set A symbolizes all the semantic correlation found with the selected algorithm.

Some of the ontology alignment techniques proposed in the literature are [8]:

- Classes Structure: Graph-based techniques that consider as input the ontologies, including database schemas and taxonomies, as labelled graphs. Usually, the similarity comparison between a pair of nodes from the two ontologies is based on the analysis of their positions within the graphs. For that, they use algorithms based on the graph theory. The intuition behind this is that, if two nodes from two ontologies are similar, their neighbors must also be somehow similar [4].

- Distance of names: These techniques consider the names of concepts as sequences of letters in an alphabet. They are typically base on the following intuition: the more similar the strings, the more likely they are to denote the same concepts. Usually, distance functions map a pair of strings to a real number, such that a smaller value indicates a greater similarity between the strings. Some examples of string-based techniques that are extensively used in matching systems are prefix or suffix distances, and n-gram similarity [4].

- Name and properties analysis: it uses the distance of the names, and merges it with another string, which describes the properties of the concepts, to compare them.

This alignment technique has been implemented in the Alignment API [2].

3 ABC Algorithm

The algorithm based on the Colonies of Bees, called Artificial Bee Colony (ABC), has been defined by [6], motivated by the intelligent behavior of bees. It is as simple as Particle Space (PSO) and Differential Evolution algorithm (DE) [6].

The ABC algorithm can be used to solve multidimensional and multimodal optimization problems. The multimodal problems have more than one maximum or minimum. In the model, artificial bee colony consists of three groups of bees: employed, onlookers and scouts. Normally, artificial bees employed compose the first half of the colony and the second half includes the onlookers and scouts. For each food source, there is only one employed bee. In other words, the number of employed bees is equals to the number of food sources around the hive. When an employee has its food source exhausted, it becomes a scout [6].

ABC has parameters as the size of the colony and the maximum number of cycles. As optimization tool, it provides a search procedure based on a population in which possible solutions represent potential sources of artificial food. The aim of the bee colony is to discover the places of food sources with high amount of nectar (good solution). In the ABC system, artificial bees fly around of a multidimensional space of search. The employed and onlookers bees choose food sources based on the experience of them and their nest mates [6].

The scout bees fly and choose food sources randomly without using the experience. If the amount of nectar from a new source is greater or better than the previous one in its memory, the new position is memorized and the previous one is forgotten.

Thus, the ABC system combines local search methods, carried out by bees employed and onlookers, which is performed through the communication of the employed bees giving their expertise to the onlookers, with the methods of global search, which are managed by the scouts when they visit randomly food sources. In this way, this technique tries to balance the exploration and exploitation process [6].

The main steps of the algorithm are [6]:

Send the scouts bees to find food sources
  REPEAT
    Send the employed bees to identified food sources and determine their amounts of nectar.

Calculate the probability value of the sources (quality) with which the onlookers bees will prefer sources.
Send onlookers bees to food sources using a stochastic selection process based on the amount of nectar in each source.
Stop the process of exploitation of sources exhausted by bees.
Send scouts to the search area to discover new food sources randomly.
Save the best food source found so far.
UNTIL (the requirements are met)

The stochastic process in the selection of food sources is given by the probability Pc. In this work, this probability allows to modify the alignment technique stochastically (change of the food source).

4 Emergent Alignment by using our ABC Approach

The problem of the ontology alignment is to be able to decide which of the techniques of semantic alignment must be used. For it, it is used the ABC algorithm in order to let it choose automatically the technique to perform the alignment.

In our problem, the parameters/variables of the ABC algorithm are the following:

- Si: Service that can be utilizes to resolve a request (In our case, the alignment techniques). That is, each alignment technique is a source of nectar.
- Gi(Si): Profit, that is obtained by the use of the service Sj (one alignment technique), defined by the equation 1, which determine the quality of nectar (alignment technique).
- Sai(Si) Satisfaction of the Bee, when the service Sj is performed. It is also relate with the quality of nectar; in our case is the number of aligned nodes of the ontologies.
- CAi(Sj): Cost, it is represented in this work as the execution time of the service Sj to return a result (also affects the quality of nectar).
- Pc: Probability of preserving the food source.

The gain is calculated as follows, in the equation (1):

$$G_i(S_i) = \frac{Sa_i(S_i)}{CA_i(S_i)} Pc \qquad (1)$$

The main part of our algorithm is the procedure followed by the onlooker bees to select the food sources, based on the experiences of the rest of bees on the colony. The scout bees memorize the best food sources that find, and the employed bees determine the food sources to be studied. The onlookers bees change the decision about which food sources exploit using the following decision rules (see figure 1)):

- For each bee i with neighbor j:

If Gi(Si) < Gj(Sj) then
Si = Sj

If the satisfaction (Sai(Si) is the same, then the service that has the shortest time is selected. It is defined as follow:

- For each bee i with neighbor j:

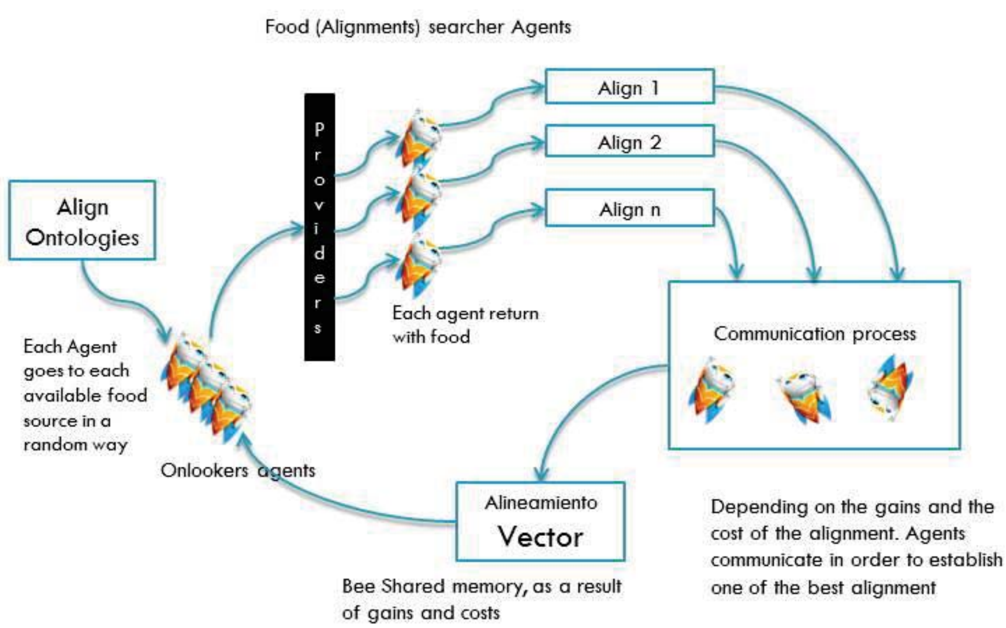

Figure 1. Emergent Alignment Process

If CAi(Si) > CAj(Sj) then
Si = Sj

The algorithm is iterative, and it is done for finite iterations to make several suggestions, it is not necessary that all the bees arrive at a same service (source of nectar); they may suggest various services. At the end of the iterations, we take the service more suggested by the bees (this is the consensus to which arrives the colony of bees).

5 Experiments

We use different pairs of ontologies to test our ABC algorithm: one pair about cars, another of the anatomy of the eye, and finally one pair about computers. These ontologies were taken from [8], where they analyze techniques to merge ontologies. Moreover, the Alignment API [2], is used to test the alignment techniques.

In Figure 2 we see the cars ontologies, which are described as taxonomies of cars, one has the brands of cars of Europe, and the other one has specifically brands of cars from German and Italy.

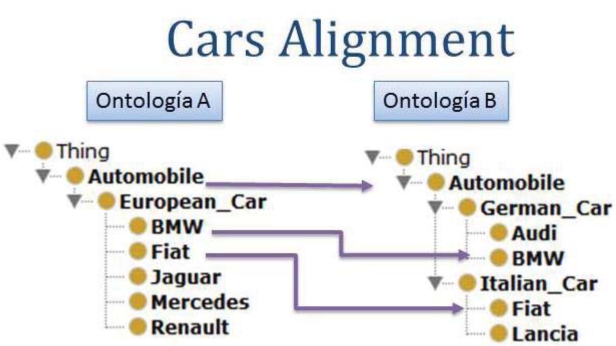

Figure 2. Ontologies to align

In Tables 1, 2, and 3, the parameters of the ABC algorithm are defined for the cars ontologies. These parameters are: the size of the colony, the maximum number of cycles (Maxcycle), the number of times that our algorithm is executed (called runtime, this to check how many times each technique is chosen as the best, among n executions of ABC). The output parameters considered for selection of the best alignment are the execution time, number of times each technique is selected, and the aligned nodes. Bold numbers in the fourth column is for indicating the best result (alignment technique).

The alignment techniques that are used by the ABC algorithm are (they determine the number of food sources, in our case 8) [8]:

a) Class Structure
b) Distance edited name
c) Distance edited subclass name
d) Name and properties
e) Same names
f) Distance SMOA name (A String Metric for Ontology Alignment) [9]
g) Distance chains
h) Sub structures distance

Table 1. Cars Ontologies. Colony size is twice the size of food sources

Maxcycle	Runtime	Execution Time	Times that each Alignment Technique is chosen	# Aligned nodes
250	30	- 6:32 6:44 6:33 6:32 6:33 6:33 7:01	a) 0 **b) 6** c) 3 d) 5 **e) 6** f) 5 g) 5 h) 0	0 3 3 3 3 3 3 3
100	30	- 2:46 2:40 2:49 2:46 2:47 2:49 2:49	a) 0 b) 4 c) 4 d) 3 **e) 6** f) 5 g) 4 h) 4	0 3 3 3 3 3 3 3
25	30	- 0:43 0:50 0:43 0:42 0:46 0:45 0:49	a) 1 b) 5 c) 1 **d) 6** **e) 6** f) 5 g) 4 h) 2	0 3 3 3 3 3 3 3

Table 2. Cars Ontologies. Colony size equal to the size of food sources

Maxcycle	Run time	Execution Time	Times that each Align Technique is chosen	# Aligned nodes
250	30	- 3:26 3:26 3:25 3:23 3:25 3:30 3:29	a) 0 b) 5 c) 5 d) 4 **e) 9** f) 5 g) 0 h) 2	0 3 3 3 3 3 3 3

100	30	-	a) 0	0
		1:26	b) 5	3
		1:24	c) 4	3
		1:22	d) 9	3
		1:23	**e) 9**	3
		1:25	f) 5	3
		1:25	g) 4	3
		1:30	h) 0	3
25	30	-	a) 1	0
		0:24	b) 6	3
		0:28	c) 3	3
		0:27	d) 2	3
		0:23	**e) 7**	3
		0:24	**f) 7**	3
		0:26	g) 2	3
		0:27	h) 2	3

Table 3. Cars Ontologies. Colony size is the half of the size of food sources

Maxcycle	**Run time**	**Execution Time**	**Times that each Align Technique is chosen**	**# Aligned nodes**
250	30	-	a) 0	0
		1:44	**b) 7**	3
		1:46	c) 5	3
		1:45	d) 4	3
		1:46	e) 4	3
		1:46	f) 5	3
		1:56	g) 3	3
		1:55	h) 2	3
100	30	-	a) 0	0
		0:49	b) 3	3
		0:44	**c) 6**	3
		0:45	d) 5	3
		0:44	e) 5	3
		0:48	f) 4	3
		0:48	g) 4	3
		0:47	h) 4	3
25	30	-	a) 1	0
		0:15	b) 5	3
		0:15	c) 4	3
		0:17	d) 4	3
		0:16	e) 4	3
		0:13	**f) 8**	3
		0:18	g) 2	3
		0:20	h) 2	3

In tables 1, 2, and 3, we show the algorithm performance in case of varying the size of the colony. This variable has important influence on the runtime and search process. It determines how many food sources can test at one time, and it requires a random search to exploit the food sources that are not been used in a given moment (it is the case of table 3). Particularly, the results show that the runtime performance can be significantly improved, without putting the amount of bees so high. Table 3 shows that with the half of the size of food sources, it is sufficient to have a good result (aligned nodes) regarding the case of having a twice as food sources (see table 1).

The number of bees should not be so large because the algorithm balances very well the process of exploration and exploitation. This is due to the ability of exploration of the technique.

Another parameter that is important to analyze is the number of cycles. We see that this parameter has a great influence regarding the techniques that are been more selected. With this parameter, the ability of the collective learning of our ABC approach is exploited in order to obtain the best techniques (b, d, e, and f). We can see that the bad technique is eliminated when we use more cycles (a). That is because the set of selected techniques changes depending on the number of cycles that the ABC algorithm leaves the colony to choose food sources. As part of the learning process the algorithm determines the set of best techniques, but with few cycles it is a random process.

The behavior of our algorithm for the rest of studied ontologies (the eye anatomy and the computers ontologies) is similar, that means the type of ontology have not influence in the quality of the search process (exploration and exploitation).

Table 4. Eye anatomy Ontologies. Case where the colony size is the half of the size of food sources for 250 cycles

Times that each Align Technique is chosen	**# Aligned nodes**
a) 0	0
b) 0	5
c) 0	5
d) 12	5
e) 0	5
f) 12	5
g) 0	5
h) 6	5

Table 5. Computer Ontologies. Case where the colony size is the half of the size of food sources for 250 cycles

Times that each Align Technique is chosen	**# Aligned nodes**
a) 0	0
b) 0	10
c) 0	10
d) 15	10
e) 0	10
f) 11	10
g) 0	10
h) 4	10

In the case of the eye anatomy ontologies, our algorithm determines the best alignment techniques are d and f, on the other hand, with computer ontologies the best alignment technique is d. We highlight they are different techniques regarding the best ones in case of the cars ontologies (normally e).

In all of cases of study, normally the analyzed techniques align the same number of concepts (except a, which is eliminated by the learning process of our ABC approach), then, the performance is evaluated in term of the execution time. In this way, the execution time of the different alignment techniques is different for each pair of ontologies to be aligned, and it is detected by our approach when selects them

6 Conclusions

As a major contribution to the field of emerging knowledge engineering, we have proposed an ABC approach to allow an automatic selection of alignment techniques. This process is adapted to the pair of ontologies that we would like to integrate for future uses. Our approach exploits very well the learning and exploration capabilities of the ABC algorithm.

This work is focuses on showing the alignment as an emergent process within an bigger process of ontology integration, when we aim to automate this process.

We have started with the assumption that there are different ways or techniques in the literature for ontology alignment, and there are no a priori criteria to decide which technique to use when we have two ontologies. We propose an equation for calculating the gain as a main criteria to propose a technique, based on the number of aligned nodes.

The gain value is used in the algorithm, and it is penalized by the execution time. Based on this value, our algorithm is capable for finding the best alignment techniques, by using the exploration and exploitation capabilities of the bees' colonies, which share information to arrive to a stable opinion, when all the bees or most of them reach a similar opinion.

Acknowledgments

This work has been developed under the funding FONACIT through the PEII program, research project No. 2994. Dr Aguilar and Dr. Cerrada have been partially supported by the Prometeo Project of the Ministry of Higher Education, Science, Technology and Innovation of the Republic of Ecuador.

7 References

[1] Alma-Delia Cuevas and Adolfo Guzman-Arenas, Automatic Fusion of knowledge stored in Ontologies. *Journal Intelligent Decision Technologies*, 4 (1): 5-19, 2010.

[2] Jérôme David, Jérôme Euzenat, François Scharffe and Cássia Trojahn dos Santos. The Alignment API 4.0, Semantic web journal 2(1):3-10, 2011

[3] Mathieu d'Aquin, Gabriel Kronberger and and María Suárez. Combining Data Mining and Ontology Engineering to Enrich Ontologies and Linked Data. In *Proceedings of the First International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data*, Greece, pages 19-24, May 2012.

[4] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, Berlin, 2013.

[5] Melanie Hilario, Phong Nguyen, Huyen Do, Adam Woznica and Alexandros Kalousis. Ontology-based meta-mining of knowledge discovery workflows. In *Meta-Learning in Computational Intelligence*, pp. 271-315, Springer, 2011.

[6] Dervis Karaboga and Brune Basturk. On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, 8: 687–697, 2008.

[7] Ammar Mechouche, Nathalie Abadie and Sébastien Mustière. Alignment-Based Measure of the Distance between Potentially Common Parts of Lightweight Ontologies. In *Proceeding of the 9th International Semantic Web Conference (ISWC'10*), Shanghai, Novembe 2010.

[8] Salvatore Raunich and Erhard Rahm, Towards a Benchmark for Ontology Merging. *Lecture Notes in Computer Science*, 7567: 124-133, 2012.

[9] Giorgos Stoilos, Giorgos Stamou and Stefanos Kollias. A String Metric for Ontology Alignment. *Lecture Notes in Computer Science*, 3729: 624-637, 2005.

[10] Hongbo Sun, Wenhui Fan, Weiming Shen; Tianyuan Xiao. Ontology fusion in HLA-based collaborative product development. 2010 In *Proceeding of International Conference on Systems Man and Cybernetics*, Turkey, pages 2526-2532, October 2010.

[11] Roberto Zagal. Alineación de ontologías usando el Método Boosting. *Tesis de grado de Maestría en Ciencias de la Computación*. Instituto Politécnico Nacional, Mexico, 2008.