

Optimizing Ontology Alignments by Using Genetic Algorithms

Jorge Martinez-Gil, Enrique Alba, and José F. Aldana-Montes

Universidad de Málaga, Departamento de Lenguajes y Ciencias de la Computación
Boulevard Louis Pasteur s/n 29071 Málaga (Spain)
{jorgemar, eat, jfam}@lcc.uma.es
<http://www.lcc.uma.es>

Abstract. In this work we present GOAL (Genetics for Ontology Alignments) a new approach to compute the optimal ontology alignment function for a given ontology input set. Although this problem could be solved by an exhaustive search when the number of similarity measures is low, our method is expected to scale better for a high number of measures. Our approach is a genetic algorithm which is able to work with several goals: maximizing the alignment precision, maximizing the alignment recall, maximizing the f-measure or reducing the number of false positives. Moreover, we test it here by combining some cutting-edge similarity measures over a standard benchmark, and the results obtained show several advantages in relation to other techniques.

Key words: ontology alignment; genetic algorithms; semantic integration

1 Introduction

The Semantic Web is a new paradigm for the Web in which the semantics of information is defined, making it possible for the web to understand and satisfy the requests of people and machines to use the web resources. Therefore, most authors consider it as a vision of the Web from the point of view of an universal medium for data, information, and knowledge exchange [1].

In relation to knowledge, it is very important the notion of ontology as a form of representation about a particular universe of discourse or some part of it. Ontology alignment is a key aspect in order to the knowledge exchange in this extension of the Web may be real; it allows organizations to model their own knowledge without having to stick to a specific standard. In fact, there are two good reasons why most organizations are not interested in working with a standard for modelling their own knowledge: (a) it is very difficult or expensive for many organizations to reach a agreement about a common standard, and (b) these standards do not often fit to the specific needs of the all participants in the standarization process.

Althought ontology alignment is perhaps the most valuable way to solve the problems of heterogeneity and, even there are a lot of techniques for aligning

ontologies in a very accurate manner, experiences tells us that the complex nature of the problem to be solved makes difficult that these techniques operate in a satisfactory way for all kinds of data, in all domains, and as all users expect. This problem has been studied in [2].

As a result, techniques that combine existing methods have appeared. The goal of these techniques is to obtain more complex and accurate matching algorithms. The way to combine these matching algorithms is under an exhaustive research now. The most promising mechanisms are reviewed in the Section 6, but we can advance that the use of Genetic Algorithms (GAs) has been studied in little depth by researchers. Therefore, the main contributions of this work are:

- The proposal of an efficient mechanism, other than those that already exist, to compute the optimal function for aligning arbitrary sets of ontologies.
- The additional possibility to obtain goal-driven results, thus optimize some of the characteristics of an output alignment.
- We provide results following a standard benchmark to enable the comparison with other approaches.

The rest of this work is structured in the following way: Section 2 describes the problem statement. Section 3 presents the technical preliminaries which are necessary to our approach. Section 4 discusses our approach. Section 5 findings extracted from several experiments, including the use of a benchmark provided by the Ontology Alignment Evaluation Initiative [3]. Section 6 compares our results with other proposals. Finally, we remark the strengths and flaws of our proposal and discuss the future work in Section 7.

2 Problem Statement

The process of aligning ontologies can be expressed as a function f where given a pair of ontologies o and o' , an partial (and optional) input alignment A , a set of parameters p and a set of resources r , returns a new alignment A' :

$$A' = f(o, o', A, p, r)$$

A' is a set of mappings. A mapping is an expression that represents a semantic correspondence between two entities. A mapping is the atomic component of an alignment and is a formalism that allows to share knowledge models created separately.

However, experience tells us that getting f is far from trivial. As we commented earlier, the heterogeneity and ambiguity of data descriptions makes unrealistic the scenario in which that optimal mappings for many pairs of entities will be considered as "best mappings" by any of the existing matching algorithms. For instance, the Fig. 1 shows an alignment that is valid for users from some countries, but not for some others. The current trend is to diversify (and possibly weight) the matching algorithms. To do it, it is necessary to use composite ontology matchers.

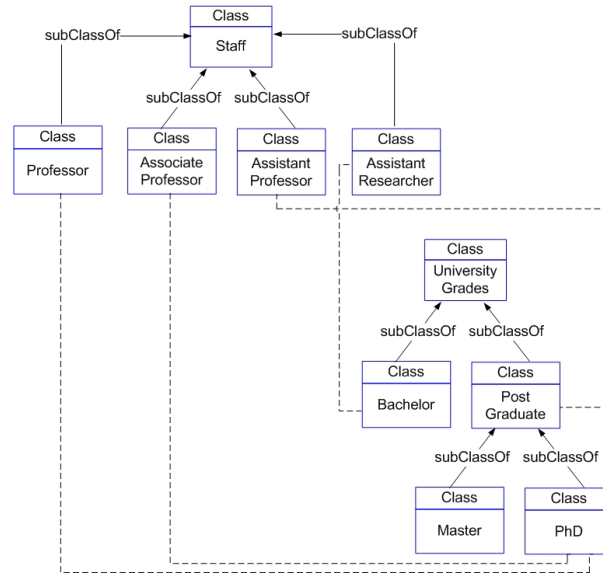


Fig. 1. Example of alignment between two ontologies. Most probably none of the two ontology owners will consider it optimal for them

Composite matchers are aggregation of simple matchers which exploit a wide range of information, in fact, we can classify the matching algorithms in the following types:

1. **String normalization.** This consists of methods such as removing unnecessary words or symbols from the entity names. Moreover, they can be used for detecting plural nouns or to take into account common prefixes or suffixes as well as other natural language features.
2. **String similarity.** Text similarity is a string based method for identifying similar entity names. For example, it may be used to identify identical concepts of two ontologies if they have a similar name. The reader can see [4] for more details about this algorithms.
3. **Data Type Comparison.** These methods compare the data type of the ontology elements. Similar concept attributes are logically expected to have the same data type.
4. **Linguistic methods.** This consists in the inclusion of linguistic resources such as lexicons and thesauri to identify possible similarities. The most popular linguistic method is to use WordNet [5] to identify some kinds of relationships between entities.
5. **Inheritance analysis.** Theses kinds of methods take into account the inheritance between concepts to identify relationships. The most popular method is the *is-a* analysis that tries to identify subsumptions between concepts.
6. **Data analysis.** These kinds of methods are based on the rule: If two concepts have the same instances, they will probably be similar. Sometimes, it

is possible to identify the meaning of an upper level entity by looking at a lower level entity. For example, if instances contain a string such as *years old*, it probably belongs to an attribute called *age*.

7. **Graph-Mapping.** This consists in identifying similar graph structures in two ontologies. These methods use known graph algorithms to do so. Most of times this involves computing and comparing paths, adjacent nodes and taxonomy leaves.
8. **Statistical analysis.** It consists of the extraction of keywords and textual descriptions for detecting the meaning of the entities in relation to other entities.
9. **Taxonomy analysis.** It tries to identify similar concepts by looking at their related concepts. The main idea is that two concepts belonging to different ontologies have a certain degree of probability of being similar if they have the same neighbours.

The main idea of composite matchers is to combine similarity values predicted by multiple simple algorithms to determine correspondences between entities belonging to different ontologies. The most popular proposals in this field are COMA [6], COMA++ [7], QuickMig [8], FOAM [9], iMAP [10] and OntoBuilder [11]. But these proposals use, in the best of the cases, weights determined by an expert. Our work does not use weights from an expert, but compute those for obtaining the optimum alignment function so that the problem can be solved accurately and without requiring human intervention.

3 Technical Preliminaries

Definition 1 (Similarity measure). A *similarity measure* sm is a function $sm : \mu_1 \times \mu_2 \mapsto \mathfrak{R}$ that associates the similarity of two input ontology entities μ_1 and μ_2 to a similarity score $sc \in \mathfrak{R}$ in the range $[0, 1]$, where a similarity score of 0 stands for complete inequality and 1 for complete equality of the input ontology entities μ_1 and μ_2 .

Definition 2 (Weighted similarity measure). Let \mathbf{A} be a set of well-known similarity measures and \mathbf{w} a numeric weight vector, and let O_1, O_2 be two input ontologies, then we can define wsm as a weighted similarity measure in the following form:

$$wsm(O_1, O_2) = x \in [0, 1] \in \mathfrak{R} \rightarrow \exists \langle \mathbf{A}, \mathbf{w} \rangle, x = \max(\sum_{i=1}^{i=n} A_i \cdot w_i)$$

subject to $\sum_{i=1}^{i=n} w_i \leq 1$

From an engineering point of view, this function leads to an optimization problem for calculating the numeric weight vector, because the number of candidates from the solution space (in this case an arbitrary continuous interval) is infinite. Hence, exact techniques are of low help here, and we are interested in methods such metaheuristics (e.g genetic algorithms) that find quasi optimum

results in such solution spaces.

Definition 3 (Ontology alignment). *An ontology alignment oa is a set of tuples $\{(id, e, e', n, R)\}$. Where id is an unique identifier of the mapping, e and e' are entities belonging to two different ontologies, R is the relation of correspondence between these entities and n is a real number between 0 and 1 that represents the mathematical probability that R is true. The entities that are related are the concepts, roles, rules, and even axioms of the two ontologies.*

Definition 4 (Ontology matching function). *An ontology matching om is a function $om : O_1 \times O_2 \xrightarrow{sm} A$ that associates two input ontologies O_1 and O_2 to an alignment A using a similarity measure (or a weighted similarity measure).*

Definition 5 (Alignment evaluation). *An alignment evaluation ae is a function $ae : A \times A_R \mapsto precision \times recall$ that associates an alignment A and an reference alignment A_R to two real numbers in the interval $[0, 1]$ stating the precision and recall of A in relation to A_R .*

Code 1 shows an example of an output from an alignment evaluation process where two ontologies from a standard benchmark provided by the OAEI [3] have been aligned. Parameters will be discussed in more detail in Section 5.

Code 1 Example of Alignment Evaluation

```
<?xml version='1.0' encoding='utf-8' standalone='yes'?>
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:map='http://.../projects/ontology/ResultsOntology.n3#'>
  <map:output rdf:about=''>
    <map:input1 rdf:resource="http://.../benchmarks/101/onto.rdf"/>
    <map:input2 rdf:resource="http://.../benchmarks/204/onto.rdf"/>
    <map:precision>1.0</map:precision>
    <map:recall>0.6288</map:recall>
    <fallout>0.0</fallout>
    <map:fMeasure>0.7721</map:fMeasure>
    <map:oMeasure>0.6288</map:oMeasure>
    <result>0.6288</result>
  </map:output>
</rdf:RDF>
```

4 Genetics for Ontology ALignments (GOAL)

We are beginning our research. First, we are going to consider GAs. Later, we may consider other approaches. GAs are often used to search along very high dimensional problems spaces. For example, if we want to find the maximum value of the function wsf with three independent variables w_0 , w_1 and w_2 :

$$wsf(O_1, O_2) = w_0 \cdot datatype(O_1, O_2) + w_1 \cdot normalization(O_1, O_2) + w_2 \cdot synonyms(O_1, O_2)$$

where w_0 , w_1 and w_2 are weights to determine the importance of the three respective similarity measures, which belong, for instance, to the continuous interval $[0, 1]$. The problem that we want to solve consists of finding a good value of w_0 , w_1 and w_2 to find the largest possible value of wsf .

While this problem can be solved trivially by a brute force search over the range of the independent variables w_0 , w_1 and w_2 , the GA method scales very well to similar problems of a higher dimensionality; for example, we might have functions using a large number of independent variables $w_0, w_1, w_2, \dots, w_n$. In this case, an exhaustive search would be prohibitively expensive.

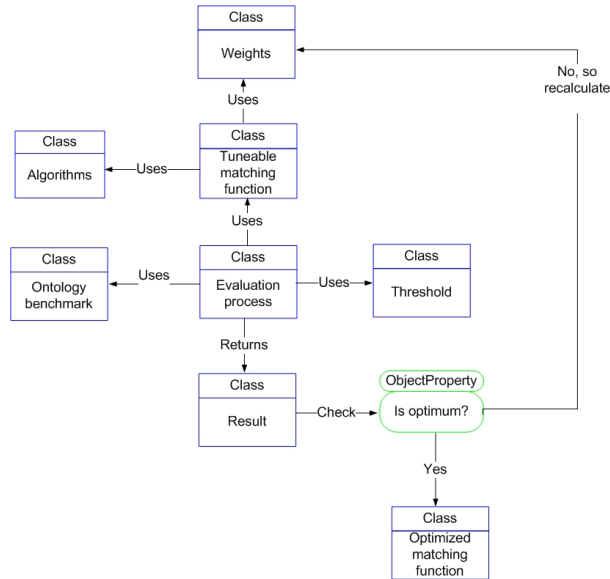


Fig. 2. General schema for our proposal

The methodology of the application of a GA requires defining the following strategies:

- Characterize the problem by encoding in a string of values the contents of a tentative solution.

- Provide a numeric fitness function that will allow to rate the relative quality of each individual tentative solution in a population.

That is what we are going to do with GOAL. Our first task is to characterize the search space as some parameters. We need to encode several parameters in a single chromosome, so we have designed a method for converting a bit representation to a set of floating-point numbers in the real range $[0, 1]$.

Later, we have designed a fitness function to determine which chromosomes in the population are most likely to survive and reproduce using genetic crossover and mutation operations.

Related to the fitness function, we can choose any parameter provided for the alignment evaluation process. In this way, we are providing the possibility to select one of these goals.

- Optimizing the precision ($fitness := precision$)
- Optimizing the recall ($fitness := recall$)
- Optimizing the f-measure ($fitness := f - measure$)
- Reducing the number of false positives ($fitness := fall - out$)

The fitness function consist of selecting one of the parameters retrieved by an Alignment Evaluation (see Definition 5). All of these parameters are concepts used in Information Retrieval [12] for measuring the quality of a retrieval task. Precision is the percentage of items returned that are relevant. Recall is the fraction of the items that are relevant to a query (in this case, to a matching task). F-measure is a harmonic mean from precision and recall. Finally, false positives are relationships which have been provided to the user although they are false. In some domains, (for instance in Medicine) false positives are absolutely unwanted.

Our algorithm works under the paradigm of a single goal programming strategy, but optimizing the F-Measure (a weighted sum of precision and recall) has an effect similar to a multi-objective strategy. However, a brief discussion about using a multi-objective algorithm will be presented as future work.

5 Empirical Evaluation

In this section, we provide an empirical evaluation of our approach. To do that, we have worked with the well-known benchmark provided by the OAEI [3]. Firstly, we have performed a preliminary study to choose the parameters and then we have performed the main experiment.

5.1 Preliminary Study

We are going to do a preliminary study of the parameters for the algorithm.

- For the number of genes per chromosome we have selected such values as 5, 10 and 20. A study using a t-Test distribution has shown us that that the differences between samples are not statistically significant. Therefore, we have selected 20 genes per chromosome.

- For the number of individuals in the population, we have selected such values as 20, 50 and 100. Again, a t-Test statistical distribution has shown that the differences between these samples are not statistically significant. So we have selected a population of 100 individuals.
- Related to crossover and mutation fraction, we have chosen a high value for the crossover between genes and, a little percentage for mutations, because we wish a classical configuration for the algorithm.
- After ten independent executions, we noticed that the genetic algorithm does not improve the results beyond the fifth generation, so we have set a limit of five generations.

5.2 Main Experiment

Related to the conditions of the experiment, we have used:

- As similarity measure vector:
 $\{Levenshtein[13], SIFO[14], Stalios[15], QGrams[16]\}$
- The GA has been configured having into account the following parameters¹:
 - 20 genes per chromosome
 - Each gene is encoded in a 10-bit representation
 - A population of 100 individuals
 - 0.98 for crossover probability
 - 0.05 for mutation probability
 - We allow 5 generations
- The platform characteristics: Intel Core 2 Duo, 2.33GHz and 4GB RAM.

The way that we have chosen for providing the dynamic evaluation of the alignment uses the following formulas:

$$Precision = \frac{\{relevant\ mappings\} \cap \{retrieved\ mappings\}}{\{retrieved\ mappings\}}$$

$$Recall = \frac{\{relevant\ mappings\} \cap \{retrieved\ mappings\}}{\{retrieved\ mappings\}}$$

$$FMeasure = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

$$Fallout = \frac{\{non\ relevant\ mappings\} \cap \{retrieved\ mappings\}}{\{non\ relevant\ mappings\}}$$

Now, let us discuss the results we have obtained. Table 1 shows a brief description about the purpose of each test of the benchmark.

Table 2 shows the results from a Precision-Driven test, the Table 3 shows the results from a Recall-Driven test, the Table 4 shows results from a F-Measure-Driven test and, finally Table 5 shows the empirical data from a Fall-out-driven test.

¹ Fitness and search space have been explained in the previous section

Ontology	Brief explanation
101	Strictly identical ontologies
102	A regular ontology and a null ontology
103	A regular ontology and other with a language generalization
104	A regular ontology and other with a language restriction
201	Ontologies without entity names
202	Ontologies without entity comments
203	Ontologies without entity names and comments
204	Ontologies with different naming conventions
205	Ontologies whose labels are synonymous
206	Ontologies whose labels are in different languages
221	A regular ontology and other with no specialisation
222	A regular ontology and other with a flatenned hierarchy
223	A regular ontology and other with a expanded hierarchy
224	Identical ontologies without instances
225	Identical ontologies without restrictions
301	A real ontology about bibliography made by MIT

Table 1. Explanation of the performed tests

Ontology	Comment	Best Precision	Generations
101	Reference alignment	1.00	1
102	Irrelevant ontology	N/A	1
103	Language generalization	1.00	1
104	Language restriction	1.00	1
201	No names	1.00	1
202	No names, no comments	1.00	1
203	No comments (was missspelling)	1.00	1
204	Naming conventions	1.00	1
205	Synonyms	1.00	2
206	Translation	1.00	2
221	No specialisation	1.00	2
222	Flatenned hierarchy	1.00	3
223	Expanded hierarchy	1.00	2
224	No instance	1.00	1
225	No restrictions	1.00	2
301	Real: BibTeX/MIT	0.90	5

Table 2. Precision-Driven test

Ontology	Comment	Best Recall	Generations
101	Reference alignment	1.00	1
102	Irrelevant ontology	N/A	1
103	Language generalization	1.00	1
104	Language restriction	1.00	1
201	No names	1.00	1
202	No names, no comments	1.00	1
203	No comments (was misspelling)	1.00	1
204	Naming conventions	1.00	1
205	Synonyms	0.71	5
206	Translation	1.00	2
221	No specialisation	1.00	1
222	Flatenned hierarchy	1.00	1
223	Expanded hierarchy	1.00	1
224	No instance	1.00	1
225	No restrictions	1.00	1
301	Real: BibTeX/MIT	0.69	5

Table 3. Recall-Driven test

Ontology	Comment	Best F-Measure (Pr, Rec)	Generat.
101	Reference alignment	1.00 (1.00, 1.00)	1
102	Irrelevant ontology	N/A	1
103	Language generalization	1.00 (1.00, 1.00)	1
104	Language restriction	1.00 (1.00, 1.00)	1
201	No names	1.00 (1.00, 1.00)	1
202	No names, no comments	1.00 (1.00, 1.00)	1
203	Comments was misspelling	1.00 (1.00, 1.00)	1
204	Naming conventions	1.00 (1.00, 1.00)	1
205	Synonyms	0.44 (0.38, 0.53)	5
206	Translation	0.43 (0.38, 0.51)	5
221	No specialisation	1.00 (1.00, 1.00)	1
222	Flatenned hierarchy	1.00 (1.00, 1.00)	2
223	Expanded hierarchy	1.00 (1.00, 1.00)	2
224	No instance	1.00 (1.00, 1.00)	3
225	No restrictions	1.00 (1.00, 1.00)	3
301	Real: BibTeX/MIT	0.57 (0.54, 0.62)	5

Table 4. F-Measure-Driven test

Ontology	Comment	Best Fallout	Generations
101	Reference alignment	0.00	1
102	Irrelevant ontology	N/A	1
103	Language generalization	0.00	1
104	Language restriction	0.00	1
201	No names	0.00	1
202	No names, no comments	0.00	1
203	No comments (was misspelling)	0.00	1
204	Naming conventions	0.00	1
205	Synonyms	0.06	5
206	Translation	0.06	5
221	No specialisation	0.00	1
222	Flatened hierarchy	0.00	2
223	Expanded hierarchy	0.00	2
224	No instance	0.00	2
225	No restrictions	0.00	3
301	Real: BibTeX/MIT	0.07	5

Table 5. Fallout-Driven test

As it can be seen, we have found the optimal alignment function for the majority of tests. In this way, we could cover matching cases, and therefore increase the chances of success. Some of test cases are solved in the first generation, this is because our application is not very difficult, maybe the problem is, but these specific instances are not.

6 Related Work

If we look at literature, we can distinguish between individual algorithms (i.e. FCA-MERGE [17] or S-Match [18]) applying only a single method of matching items i.e. linguistic or taxonomical matchers and combinations of the former ones, which intend to overcome their limitations by proposing hybrid and composite solutions. A hybrid approach (i.e. Cupid [19]) follows a black box paradigm, in which various individual matchers are melt together in a new algorithm [20], while the so-called composite matchers allow an increased user interaction (i.e. COMA++ [7], Falcon [21], CtxMatch [22], RiMOM [23]). In Fig. 3, we can see a comparison between some of the most popular tools for matching ontologies. The figure represents the arithmetic means of the values obtained for the standard benchmark for the precision and recall, obtaining the F-Measure and Fall-Out is trivial.

The problem is that those kinds of proposals use weights defined by an expert for configuring the composite matchers, while using our approach involves to compute the weights in an automatic way, so the process can be more flexible, at least, in real scenarios.

To avoid the expert intervention, there are two research lines; one line for evaluating the results of an alignment tool and maybe feedback the process

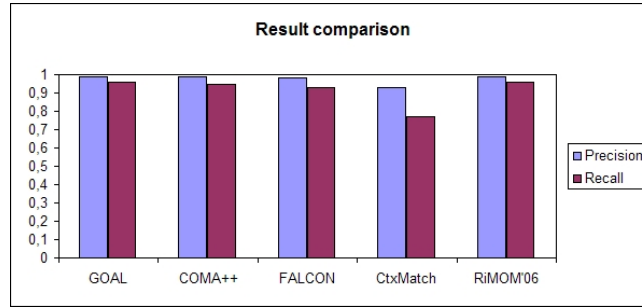


Fig. 3. Comparison between most outstanding tools

[24] [25] and another called ontology meta-matching [26] that tries to optimize automatically the parameters related to matching task. So, our approach could be considered a mechanism for meta-matching. Most outstanding examples for this paradigm are evaluated in the next sections: (i) Exhaustive Search solutions, (ii) Machine Learning solutions, and (iii) Genetic Algorithms solutions.

6.1 Exhaustive Search

Ontology meta-matching can be solved trivially by an exhaustive search when the number of similarity measures is low. The most popular approach in this sense is eTuner [27] that it is a system which, given a particular matching task, automatically tunes an ontology matching system (computing one-to-one alignments). For that purpose, it chooses the most effective basic matchers, and the best parameters to be used.

However, exhaustive searches are very expensive, and unworkable when combining a great number of measures, from a computational point of view. Unfortunately, the paper from eTuner [27] has not used an standard benchmark to offer the results, so we cannot show a comparison.

6.2 Machine Learning

Based on Machine Learning meta-matching techniques can be divided into two subtypes: Relevance feedback [28] and Neural Networks [29]:

- The idea behind relevance feedback [28] is to take the results that are initially returned from a given query and to use information about whether or not those results are relevant to perform a new query: APFEL (Alignment Process Feature Estimation and Learning) [29] is a machine learning approach that explores user validation of initial alignments for optimising automatically the configuration parameters of some of the matching strategies of the system, e.g., weights, and thresholds, for the given matching task.

- Neural Networks [30] are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. SFS [31] It is a tool for ontology meta-matching that tries to obtain automatically a vector of weights for different semantic aspects of a matching task, such as comparison of concept names, comparison of concept properties, and comparison of concept relationships. To do that, it uses neural networks.

However, these kind of solutions implies spending much time on training the systems in relation to our proposal.

6.3 Genetic Algorithms

In relation to other based-on-Genetic-Algorithm solutions, the most outstanding tool is GAOM [32] which is a genetic algorithm based approach for solving the ontology matching problem. For the purpose of a more precise representation of ontology features, it defines two aspects: intensional and extensional. On the other hand, ontology matching problem is modeled as a global optimization of a mapping between two ontologies. Then, a genetic algorithm is used to achieve a quasi optimal solution.

Table 7 shows a comparison of the results we have obtained for both GAOM and GOAL.

	Precision	Recall
GAOM	0.94	0.87
GOAL	0.99	0.96

Table 6. Comparison between GAOM and our proposal

Although we also follow a GA based paradigm, our GOAL is slightly better in terms of numbers to GAOM as our results shows. We think that the main difference in relation to the other tool is the fitness function. Therefore, as far as we know, our results constitute the new state of the art (S.O.T.A.) in this domain.

7 Conclusions and Future Work

We have here presented a mechanism for obtaining optimum ontology alignment functions using genetic algorithms which is part of a novel computational discipline, called meta-matching, which allows flexible and accurate automatic ontology matching and generalizes and extends previous proposals for exploiting an ensemble of ontology matchers.

We have shown that our proposal is able to find the optimal solutions for ontology alignment in most cases. According to the results, our approach seems

to be an accurate and efficient tool for this task. And most importantly, it can be used in a single goal-driven way versus others using composite matching algorithms.

However, this mechanism is heavily dependent of the similarity measures to be weighted. By this reason, we highly recommend to use not only cutting-edge measures, but a big enough and representative set of them. We recommend to use, at least, one similarity measure for each kind of the matchers discussed in Section 2.

As future work, we want to study a multiobjective strategy, thus, we plan to avoid unwanted deviations from precision and recall values. Moreover, we want to learn more about [33] for automatically selecting matching algorithms on the basis of their metadata. Our goal is, given the specifications of an ontology matching problem, to compute the optimum alignment function so that the problem can be solved accurately and without requiring human intervention. In this way, the real interoperability in the Semantic Web might become true.

Acknowledgements

This work has been funded by Spanish Ministry of Education & Science through the research projects: *A basic infrastructure for development in the Semantic Web and its application to conceptual mediation in Bioinformatics*. (TIN2005-09098-C05-01) and *Net Centric Optimization*. (TIN2005-08818-C04-01).

References

1. Tim Berners-Lee, James Hendler and Ora Lassila. The Semantic Web. Scientific American, May 2001.
2. Christoph Kiefer, Abraham Bernstein, Markus Stocker: The Fundamentals of iSPARQL: A Virtual Triple Approach for Similarity-Based Semantic Web Tasks. ISWC/ASWC 2007: 295-309.
3. Ontology Alignment Evaluation Initiative (OAEI). <http://oaei.ontologymatching.org/2007>. Last visit: 29-jan-2008.
4. Gonzalo Navarro: A guided tour to approximate string matching. ACM Comput. Surv. 33(1): 31-88 (2001).
5. WordNet. <http://wordnet.princeton.edu>. Visit date: 11-march-2008.
6. Hong Hai Do, Erhard Rahm: COMA - A System for Flexible Combination of Schema Matching Approaches. VLDB 2002: 610-621.
7. David Aumüller, Hong Hai Do, Sabine Massmann, Erhard Rahm: Schema and ontology matching with COMA++. SIGMOD Conference 2005: 906-908.
8. Christian Drumm, Matthias Schmitt, Hong Hai Do, Erhard Rahm: Quickmig: automatic schema matching for data migration projects. CIKM 2007: 107-116.
9. Marc Ehrig, York Sure: FOAM - Framework for Ontology Alignment and Mapping - Results of the Ontology Alignment Evaluation Initiative. Integrating Ontologies 2005.
10. Robin Dhamankar, Yoonkyong Lee, AnHai Doan, Alon Y. Halevy, Pedro Domingos: iMAP: Discovering Complex Mappings between Database Schemas. SIGMOD Conference 2004: 383-394.

11. Haggai Roitman, Avigdor Gal: OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources Using Sequence Semantics. EDBT Workshops 2006: 573-576.
12. Ricardo A. Baeza-Yates, Berthier A. Ribeiro-Neto: Modern Information Retrieval ACM Press / Addison-Wesley 1999.
13. Vladimir Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics-Doklady*, Vol. 10, pages 707-710, August 1966.
14. Jorge Martinez-Gil, Ismael Navas-Delgado, Jose F. Aldana Montes. SIFO. An efficient taxonomical matcher for ontology alignment. Technical Report ITI-08-3. Department of Languages and Computing Sciences, University of Malaga. February 2008.
15. Giorgos Stoilos, Giorgos B. Stamou, Stefanos D. Kollias: A String Metric for Ontology Alignment. International Semantic Web Conference 2005: 624-637
16. Esko Ukkonen: Approximate String Matching with q-grams and Maximal Matches. *Theor. Comput. Sci.* 92(1): 191-211 (1992).
17. Gerd Stumme, Alexander Maedche: FCA-MERGE: Bottom-Up Merging of Ontologies. IJCAI 2001: 225-234.
18. Fausto Giunchiglia, Pavel Shvaiko, Mikalai Yatskevich: S-Match: an Algorithm and an Implementation of Semantic Matching. ESWS 2004: 61-75.
19. Jayant Madhavan, Philip A. Bernstein, Erhard Rahm: Generic Schema Matching with Cupid. VLDB 2001: 49-58.
20. Carmel Domshlak, Avigdor Gal, Haggai Roitman: Rank Aggregation for Automatic Schema Matching. *IEEE Trans. Knowl. Data Eng.* 19(4): 538-553 (2007).
21. Wei Hu, Gong Cheng, Dongdong Zheng, Xinyu Zhong, Yuzhong Qu: The Results of Falcon-AO in the OAEI 2006 Campaign. *Ontology Matching 2006*.
22. Slawomir Niedbala: OWL-CtxMatch in the OAEI 2006 Alignment Contest. *Ontology Matching 2006*.
23. Yi Li, Juan-Zi Li, Duo Zhang, Jie Tang: Result of Ontology Alignment with Ri-MOM at OAEI'06. *Ontology Matching 2006*.
24. Avigdor Gal, Ateret Anaby-Tavor, Alberto Trombetta, Danilo Montes: A framework for modeling and evaluating automatic semantic reconciliation. *VLDB J.* 14(1): 50-67 (2005).
25. Patrick Lambrix, He Tan: A Tool for Evaluating Ontology Alignment Strategies. *J. Data Semantics* 8: 182-202 (2007)
26. Jerome Euzenat, Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, 2007.
27. Yoonkyong Lee, Mayssam Sayyadian, AnHai Doan, Arnon Rosenthal: eTuner: tuning schema matching software using synthetic scenarios. *VLDB J.* 16(1): 97-122 (2007).
28. Gerard Salton, Chris Buckley: Improving retrieval performance by relevance feedback. *JASIS* 41(4):288-297 (1990).
29. Marc Ehrig, Steffen Staab, York Sure: Bootstrapping Ontology Alignment Methods with APFEL. International Semantic Web Conference 2005: 186-200.
30. Michael I. Jordan, Christopher M. Bishop: *Neural Networks*. The Computer Science and Engineering Handbook 1997: 536-556.
31. Jingshan Huang, Jiangbo Dang, Jos M. Vidal, and Michael N. Huhns. Ontology Matching Using an Artificial Neural Network to Learn Weights. IJCAI Workshop on Semantic Web for Collaborative Knowledge Acquisition 2007.
32. J. Wang, Z. Ding, C. Jiang: GAOM: Genetic Algorithm based Ontology Matching. In Proceedings of IEEE Asia-Pacific Conference on Services Computing, 2006.
33. Malgorzata Mochol, Elena Paslaru Bontas Simperl: A High-Level Architecture of a Metadata-based Ontology Matching Framework. DEXA Workshops 2006: 354-358