A genetic algorithms-based approach for optimizing similarity aggregation in ontology matching

Marcos Martínez-Romero[1], José Manuel Vázquez-Naya[2], Francisco Javier Nóvoa[2], Guillermo Vázquez[3], and Javier Pereira[1]

[1] *IMEDIR Center, University of A Coruña, Campus de Elviña s/n, 15071, A Coruña, Spain*
[2] *Department of Information and Communication Technologies, Computer Science Faculty, University of A Coruña, 15071, A Coruña, Spain*
[3] *Institute of Biomedical Research of A Coruña (INIBIC), Xubias de Arriba 84, Hospital Materno Infantil (1ª planta), 15006 , A Coruña, Spain*

Abstract
Ontology matching consists of finding the semantic relations between different ontologies and is widely recognized as an essential process to achieve an adequate interoperability between people, systems or organizations that use different, overlapping ontologies to represent the same knowledge. There are several techniques to measure the semantic similarity of elements from separate ontologies, which must be adequately combined in order to obtain precise and complete results. Nevertheless, combining multiple similarity measures into a single metric is a complex problem, which has been traditionally solved using weights determined manually by an expert, or through general methods that do not provide optimal results. In this paper, a genetic algorithms based approach to aggregate different similarity metrics into a single function is presented. Starting from an initial population of individuals, each one representing a combination of similarity measures, our approach allows to find the combination that provides the optimal matching quality.

Keywords
Genetic algorithms; Ontology matching; Ontologies; Semantic Web

1. Introduction

At present, the role of ontologies as the essential artifact for allowing a more effective data and knowledge sharing and reusing in the Semantic Web [1] is widely recognized [2] and a variety of public ontologies exist for different areas. This innovative knowledge representation method is considered to be an appropriate solution to the problem of heterogeneity in data, since ontological methods make it possible to reach a common understanding of concepts in a particular domain, supporting the exchange of information between people (or systems) that utilize different representations for the same or similar knowledge [3, 4].

Nevertheless, given that different tasks or different points of view usually require different conceptualizations, utilizing a single ontology is neither always possible nor advisable. This can lead to the usage of different ontologies, although in some cases they might contain information that could be overlapping. This, in turn, represents another type of heterogeneity that can result in inefficient processing or misinterpretation of data, information, and knowledge. Addressing this problem requires to find the correspondences, or mappings, that exist between the elements of the different ontologies being used. This process is commonly known as *ontology matching, mapping or alignment* [5]. The resulting set of inter-ontology relations can be used to adequately exchange information between people, systems and organizations.

During the last years, multiple *ontology alignment techniques* have been conceived to identify these correspondences [6]. These methods are based on computing a similarity (or distance) value between elements of different ontologies. When computing the ontology alignment between two ontologies, it is frequently to use several ontology alignment techniques, based on different similarity approaches (e.g. lexical similarity, structural similarity, etc.) and then aggregating them into a unique similarity value. However, calculating the optimal similarity aggregation is a computationally expensive task that requires new, more efficient methods to get precise and complete alignments [5, 7, 8].

In this work, we propose an approach based on genetic algorithms (GAs) to ascertain how to combine multiple similarity measures into a single aggregated metric, in order to provide the optimal matching result. Our work can be useful to automatically tune an ontology matching system in environments where a reference matching is provided.

Qazvinian et al. [9] are among the small number of authors who have tried, up to the moment, to apply GAs to the ontology matching task. They considered ontology matching as an optimization problem in which the objective is maximizing the overall similarity value between the input ontologies, and they used a GA to find the optimal mapping. In a similar way, another interesting approach to ontology matching by using GAs is GAOM [10]. In this work, ontology features are defined from two aspects: intensional and extensional, and the ontology matching problem is modeled as a global optimization of a mapping between two ontologies. Then GAs are used to achieve an approximate optimal solution.

2. A New Approach to Optimize Similarity Aggregation

In this section, a genetic algorithm to find the optimal aggregation of multiple similarity measures is presented. The GA starts from a randomly generated aggregation of similarity measures (set of weights), and tries to find the weights that optimize the global matching quality. In order to reliably describe the proposed strategy, it is necessary to define the following elements (see Fig. 1):

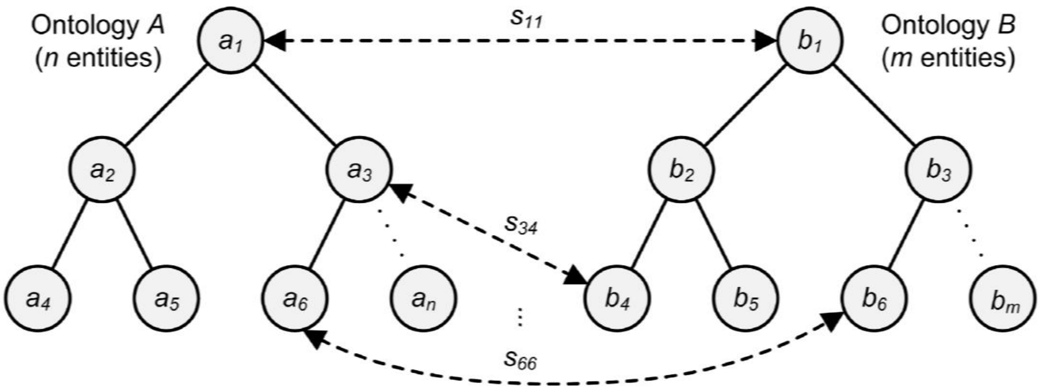

Fig. 1. Graphical representation of the ontology matching problem. The figure shows the taxonomy of two ontologies (*A* and *B*), and a set of semantic mappings (s_{ij}) between them.

- *A* and *B* are two ontologies with *n* and *m* elements (entities) respectively. *A* is composed by the entities $a_1,...,a_n$ while B has the entities $b_1,..., b_m$.

- *S* is an existing set of semantic mappings or correspondences s_{ij} between *A* and *B*, being s_{ij} a semantic mapping between the entity a_i from *A* and the entity b_j from *B*, with $0 < i \leq n$ and $0 < j \leq m$.

- $F = \{F_1(a_i,b_j),...,F_p(a_i,b_j)\} = \{F_1(s_{ij}),...,F_p(s_{ij})\}$ is a set composed by *p* functions, or ontology matching metrics, to compute a value of semantic similarity (in the [0, 1] interval) between pairs of entities from separate ontologies.

- *t* is a similarity threshold belonging to the interval [0, 1], which indicates the minimum similarity value required to consider that exists a semantic correspondence between two different entities.

- $F_{agg}(a_i, b_j) = \sum_{k=1}^{p} w_k \cdot F_k(a_i, b_j)$, with $\sum_{k=1}^{p} w_k = 1$, is a function to compute an aggregated similarity value between two entities. This function combines the similarity values provided by *p* different similarity functions into a single value belonging to the interval [0, 1]. The aggregation is based on the values of a set of *p* weights w_k, which quantify the contribution of each separate similarity measure to the aggregated value.

- $Q(S) \rightarrow [0,1]$ is a function that measures the quality of a set of semantic correspondences between two ontologies. A good example of quality measure is the f-measure metric, which considers both the precision and the recall to compute the score.

The approach is addressed to find the values of the weights w_k that maximize the quality of the matching between the input ontologies *A* and *B*, that is, the function *Q(S)*. The obtained set of weights could be subsequently used to compute the matching of ontologies with similar characteristics, or belonging to the same domain as the ontologies whose matching was selected as a reference.

2.1 Encoding Mechanism and Initialization

Each individual in the population represents a potential solution to the problem, that is, a set of weights w_k that indicate the contribution of each similarity metric to the aggregated similarity function. We propose an encoding mechanism based on that each position in the chromosome contains a value in the interval [0, 1], which represents a *cut*, or *separation point* that limits the value of a weight (remember that the summation of all weights is equal to 1). Considering that p is the number of required weights, the set of cuts could be formally represented as $C' = \{c_{1'}, \ldots, c_{p-1'}\}$. The chromosome decoding is carried out by ordering C' from lower to higher, which constitutes the ordered set of values $C = \{c_1, \ldots, c_{p-1}\}$, and calculating the weights according to the following expression:

$$w_k = \begin{cases} c_1 & k = 1 \\ c_i - c_{i-1}, & l < k < p \\ l - c_{p-1}, & k = p \end{cases}$$

A graphical representation of the chromosome and the decoded values is presented in Fig. 2, while Fig. 3 shows an example that can be useful to understand the decoding mechanism.

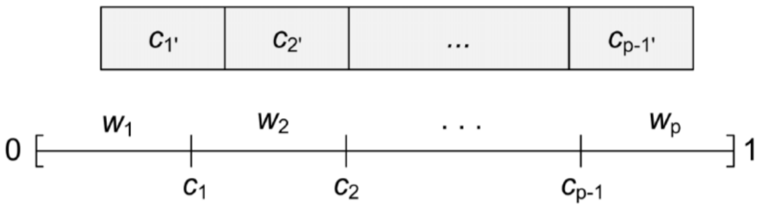

Fig. 2. Graphical representation of a chromosome and the set of weights obtained after decoding it. Each gene in the chromosome contains a value belonging to the interval [0, 1] that represents a cut, or separation point between weights. $C' = \{c_{1'}, \ldots, c_{p-1'}\}$ is an unordered set of cuts, while $C = \{c_1, \ldots, c_{p-1}\}$ is the result obtained after ordering C' from lower to higher. $W = \{w_1, \ldots, w_p\}$ is the set of weights that constitute the solution to the problem.

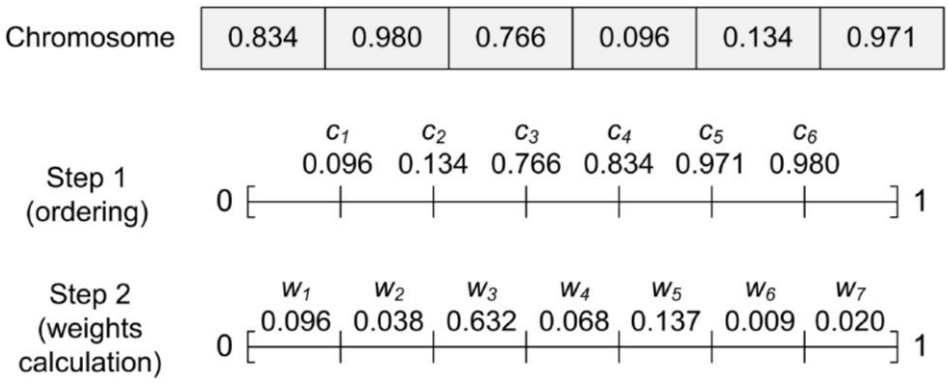

Fig. 3. Example of a specific individual and the weights obtained after decoding it. In this example, 7 different weights were considered.

2.2 Reproduction Methods

To go from one generation to the next one, we suggest using the following operators:

- **Selection.** We propose to use a roulette wheel selection method, which consists in that individuals are given a probability of being selected that is directly proportionate to their fitness, so the best individuals will have more opportunities of reproduction. Two individuals are then chosen randomly based on these probabilities and produce offspring.
- **Crossover.** Crossover will use a non-destructive strategy, in such a way that the descendants will pass to the following generation only if they exceed the fitness of their parents. A single-point crossover will be used, which consists in randomly selecting a crossover point on both parent chromosomes and then interchanging the two parent chromosomes to produce two new offspring.
- **Copy.** The best individual from one generation will be also copied to the following generation (elitist strategy). This decision has been taken to keep the best set of weights (best solution) that has been obtained up to the moment**.**
- **Mutation.** When the crossover has been achieved, genes will be mutated with a low probability. This mutation will consist in replacing the selected gene by a randomly generated one.

2.3 Fitness Function

For the smooth running of a GA, it is necessary to have a method that allows to show if the individuals of the population are or are not good solutions to the problem. That is the aim of the fitness or objective function. As our fitness function, we propose to use the *f-measure* [11], which is the uniformly weighted harmonic mean of precision and recall. F-measure will be used as the reference quality metric, in such a way that we will consider that the best alignment is the alignment with highest f-measure.

$$fitness = f - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

2.4 Stop Criterion

We propose to use a hybrid stop criterion: the GA will stop when one of the following conditions is true: (1) A fixed number of iterations have been reached; (2) The value for the fitness function is higher than a particular threshold.

3 Execution Example

In this section we provide a “toy” example with two small ontologies, which can be useful to understand how the proposed GA works. We will assume that:

- A and B are two ontologies from a specific domain. Both ontology A and ontology B have 3 entities (n = 3, m = 3).
- S is the reference matching between A and B. In this example $S = \{s_{12}, s_{33}\}$, that is, it will be supposed that there is a semantic mapping between the pairs of entities (a_1, b_2) and (a_3, b_3), as shown in Fig. 4. We will also suppose that there are some similarity between the entities (a_1, b_1) and (a_2, b_2) , but not enough to be considered semantic mappings.
- $F = \{F_1(s_{ij}), F_1(s_{ij}), F_2(s_{ij}), F_3(s_{ij}), F_4(s_{ij}), F5(s_{ij})\}$is a set composed by fivedifferent similarity functions. We need to aggregate the similarity valuesprovided by these functions into a single measure. We will also suppose that thefunctions $F_3(s_{ij})$ and $F_5(s_{ij})$,due to the particular characteristics of A and B, arenot adequate to align them, so they will not provide reliable similarity values.

- The similarity threshold t is set to 0.7, which means that the algorithm will consider that exists a mapping between a pair of entities (a_i, b_j) if the similarity function for such entities provides a value higher than 0.7. We will also suppose that the algorithm will finish if the value of the fitness function is higher than 0.8 (stop criterion).

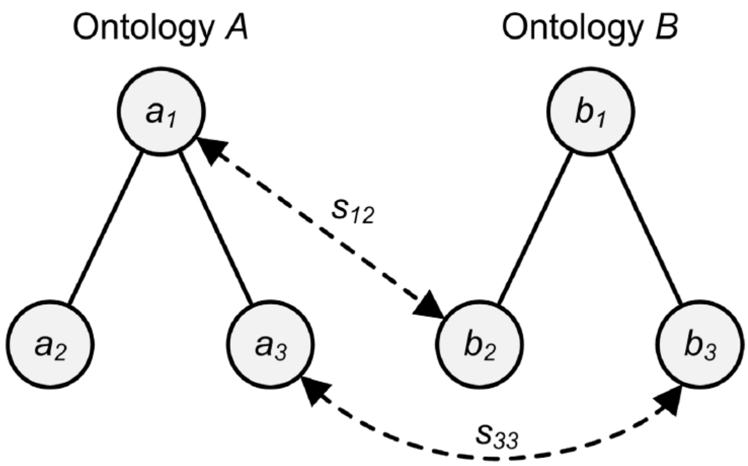

Fig. 4. Graphical view of ontologies *A* and *B*, and the reference matching *S* (gold standard)

Considering the previous information, the aggregated similarity function would be: $F_{agg}(a_i,b_j) = w_1 \cdot F_1(s_{ij}) + w_2 \cdot F_2(s_{ij}) + w_3 \cdot F_3(s_{ij}) + w_4 \cdot F_4(s_{ij}) + w_5 \cdot F_5(s_{ij})$, with $w_1 + w_2 + w_3 + w_4 + w_5 = 1$. The GA will be used to find the values of weights w_1, w_2, w_3, w_4 and w_5 that provide the optimal matching.

Firstly, it is necessary to compute the values of similarity for the n x m possible correspondences between *A* and *B*, according to the five different similarity functions. In this example, we will suppose that the results of this computation are the ones in Table 1. Remember that it has been supposed that the correct mappings are s_{12} and s_{33}, and that $F_3(s_{ij})$ and $F_5(s_{ij})$ are not adequate to align the given ontologies, so they are not able to identify s_{12} and s_{33} as the valid mappings.

Table 1. Results of initial similarity computation

	s_{11}	s_{12}	s_{13}	s_{21}	s_{22}	s_{23}	s_{31}	s_{32}	s_{33}
$F_1(s_{ij})$	0.56	0.93	0.12	0.05	0.66	0.31	0.08	0.18	0.97
$F_2(s_{ij})$	0.65	0.99	0.20	0.03	0.68	0.49	0.03	0.23	0.81
$F_3(s_{ij})$	0.11	0.17	0.23	0.41	0.56	0.11	0.65	0.09	0.21
$F_4(s_{ij})$	0.72	0.72	0.44	0.50	0.45	0.11	0.01	0.13	0.98
$F_5(s_{ij})$	0.77	0.28	0.81	0.74	0.98	0.79	0.87	0.17	0.09

The following step would be to generate the initial population. In this case, it is composed by 10 randomly generated individuals, which are shown in Table 2.

Table 2. Initial population (first generation)

Individual	Values				Individual	Values			
1	0.37	0.62	0.23	0.43	6	0.30	0.27	0.92	0.71
2	0.32	0.08	0.07	0.56	7	0.69	0.22	0.17	0.94
3	0.53	0.91	0.11	0.73	8	0.22	0.66	0.45	0.21
4	0.65	0.63	0.01	0.70	9	0.20	0.14	0.25	0.12
5	0.86	0.19	0.59	0.21	10	0.85	0.53	0.41	0.19

The next step would be to calculate the fitness value for each individual. Each chromosome is decoded in order to obtain the values for the 5 weights (see Table 3).

Table 3. Weights for the 1st generation, obtained after decoding the chromosomes in Table 2

Individual	w_1	w_2	w_3	w_4	w_5
1	0.23	0.14	0.06	0.19	0.38
2	0.07	0.01	0.24	0.24	0.44
3	0.11	0.42	0.20	0.18	0.09
4	0.01	0.62	0.02	0.05	0.30
5	0.19	0.02	0.38	0.27	0.14
6	0.27	0.03	0.41	0.21	0.08
7	0.17	0.05	0.47	0.25	0.06
8	0.21	0.01	0.23	0.21	0.34
9	0.12	0.02	0.06	0.05	0.75
10	0.19	0.22	0.12	0.32	0.15

The obtained weights are then used to compute the aggregated similarity value for each possible correspondence. These values are shown in Table 4. As an example, the aggregated value for the correspondence (a_1, b_1) and the weights obtained after decoding the individual 1, would be calculated as:

$$\begin{aligned} F_{agg}(a_1, b_1) &= w_1 \cdot F_1(s_{11}) + w_2 \cdot F_2(s_{11}) + w_3 \cdot F_3(s_{11}) + w_4 \cdot F_4(s_{11}) + w_5 \cdot F_5(s_{11}) \\ &= 0.23 \cdot 0.56 + 0.14 \cdot 0.65 + 0.06 \cdot 0.11 + 0.19 \cdot 0.72 + 0.38 \cdot 0.77 = 0.66 \end{aligned}$$

Table 4. Aggregated similarity values for the initial population. The table also shows the mappings that exceed the similarity threshold (0.70), which are used to calculate the fitness value for each individual.

Ind.	F_{agg11}	F_{agg12}	F_{agg13}	F_{agg21}	F_{agg22}	F_{agg23}	F_{agg31}	F_{agg32}	F_{agg33}	Mappings	Fitness
1	0.66	0.61	0.46	0.42	**0.74**	0.47	0.39	0.17	0.57	s_{22}	-
2	0.58	0.41	0.53	0.55	**0.73**	0.43	0.55	0.14	0.40	s_{22}	-
3	0.56	**0.71**	0.30	0.26	0.64	0.35	0.23	0.17	0.67	s_{12}	0.67
4	0.68	**0.75**	0.39	0.27	**0.76**	0.55	0.29	0.20	0.59	s_{12}, s_{22}	0.50
5	0.46	0.49	0.35	0.40	0.61	0.25	0.39	0.13	0.56	-	-
6	0.43	0.52	0.29	0.35	0.60	0.23	0.36	0.13	0.59	-	-
7	0.41	0.48	0.30	0.37	0.58	0.20	0.38	0.13	0.55	-	-
8	0.56	0.49	0.45	0.46	0.70	0.39	0.46	0.15	0.50	-	-
9	0.70	0.39	0.66	0.61	**0.88**	0.65	0.70	0.17	0.26	s_{22}	-
10	0.61	0.69	0.36	0.34	0.63	0.33	0.23	0.16	**0.71**	s_{33}	0.67

The correspondences with a similarity value higher than the given threshold (0.7) are considered valid semantic mappings. Using these mappings and the reference matching (gold standard), the fitness value (f-measure) is calculated. There are two individuals (3 and 10) that provide a fitness value of 0.67, but this value is not enough to stop the algorithm according to the fitness threshold that has been set (0.8). As a consequence, the next step is to select the individuals that will reproduce themselves to create the next generation.

The individuals that form the second generation are shown in Table 5. According to an elitist strategy, the individuals 3 and 10 are copied to the second generation (they are named 11 and 12). We suppose that the roulette selection method selects the individuals 3 and 10 to reproduce themselves and that a single-point crossover is applied between genes 1 and 2, giving as a result individuals 13 and 14; in the middle point (individuals 15 and 16); and between genes 3 and 4 (individuals 17 and 18). Individuals 19 and 20 are obtained by mutating one gene from the individuals 3 (gene 1) and 10 (gene 2), respectively. The corresponding weights are shown in Table 6.

Table 5. Second generation

Individual	Values				Individual	Values			
11	0.53	0.91	0.11	0.73	16	0.85	0.53	0.11	0.73
12	0.85	0.53	0.41	0.19	17	0.53	0.91	0.11	0.19
13	0.53	0.53	0.41	0.19	18	0.85	0.53	0.41	0.73
14	0.85	0.91	0.11	0.73	19	0.25	0.91	0.11	0.73
15	0.53	0.91	0.41	0.19	20	0.85	0.87	0.41	0.19

Table 6. Weights for the 2nd generation, obtained after decoding the chromosomes in Table 5

Individual	w_1	w_2	w_3	w_4	w_5
11	0.11	0.42	0.20	0.18	0.09
12	0.19	0.22	0.12	0.32	0.15
13	0.19	0.22	0.12	0.00	0.47
14	0.11	0.62	0.12	0.06	0.09
15	0.19	0.22	0.12	0.38	0.09
16	0.11	0.42	0.20	0.12	0.15
17	0.11	0.08	0.34	0.38	0.09
18	0.41	0.12	0.20	0.12	0.15
19	0.11	0.14	0.48	0.18	0.09
20	0.19	0.22	0.44	0.02	0.13

Table 7. Aggregated similarity values, mappings and fitness for the second generation

Ind.	F_{agg11}	F_{agg12}	F_{agg13}	F_{agg21}	F_{agg22}	F_{agg23}	F_{agg31}	F_{agg32}	F_{agg33}	Mappings	Fitness
11	0.56	**0.71**	0.30	0.26	0.64	0.35	0.23	0.17	0.67	s_{12}	0.67
12	0.61	0.69	0.36	0.34	0.63	0.33	0.23	0.16	**0.71**	s_{33}	0.67
13	0.62	0.55	0.48	0.41	**0.80**	0.55	0.51	0.18	0.43	s_{22}	-
14	0.59	**0.80**	0.26	0.17	0.68	0.43	0.18	0.20	0.70	s_{12}	0.67
15	0.61	**0.71**	0.33	0.32	0.60	0.29	0.18	0.16	**0.77**	s_{12}, s_{33}	1
16	0.56	0.68	0.32	0.27	0.67	0.39	0.28	0.18	0.62	-	-
17	0.49	0.54	0.35	0.40	0.58	0.22	0.31	0.13	0.62	-	-
18	0.53	0.66	0.29	0.28	0.67	0.34	0.30	0.16	0.67	-	-
19	0.40	0.48	0.30	0.36	0.61	0.25	0.41	0.13	0.51	-	-
20	0.41	0.52	0.28	0.30	0.66	0.32	0.42	0.15	0.49	-	-

The aggregated similarity values for the second generation are shown in Table 7. It is possible to see that the individual 15 has a fitness value of 1, which is the maximum value for the fitness function. Having reached this value, the GA stops (according to the stop criterion). The GA has provided the following solution to the problem, obtained after decoding the individual 15:

$$w_1 = 0.19;\ w_2 = 0.22;\ w_3 = 0.12;\ w_4 = 0.38;\ w_5 = 0.09$$

Given these weights, the aggregated similarity function for this example would be calculated according the following expression:

$$F_{agg}(s_{ij}) = 0.19 \cdot F_1(s_{ij}) + 0.22 \cdot F_2(s_{ij}) + 0.12 \cdot F_3(s_{ij}) + 0.38 \cdot F_4(s_{ij}) + 0.09 \cdot F_5(s_{ij})$$

As it can be observed, this function gives a low weight to the functions $F_3(s_{ij})$ and F 5(sij). We had supposed that F_3 and F_5 were not reliable, so the result provided by the approach makes sense. Using this function, we could align any pair of ontologies with similar characteristics to *A* and *B*.

4 Conclusion and Future Research

Although a lot has been done towards tackling ontology matching, the research community still reports open issues that impose new challenges for researchers and underline new directions for the future. One of these issues, which represents an emerging research area, is the aggregation of different similarity measures into a single one. In this work, we have proposed a GA-based approach to combine different measures into a single metric, optimizing the quality of the matching results. The presented GA can be useful to automatically configure the similarity aggregation process in ontology matching systems addressed to provide precise and complete results in domains that require rapid processing. Through a simple example, we have showed how the GA can find the similarity combination that provides an optimal matching result between two ontologies.

The most immediate future work is to embed our GA into a real existing ontology matching system that achieves similarity aggregation in a traditional manner (i.e., either through manual, user-based aggregation or by means of general methods), in order to measure the improvement of matching quality. We are also interested in extending our theory and mechanisms for providing an ontology matching system with full self-configuration capabilities, in order to obtain good results in dynamic environments that require immediate response, without requiring user interaction.

Acknowledgements. Work supported by the Carlos III Health Institute (grant FISPI10/ 02180), the Ibero-NBIC Network (ref. 209RT0366) funded by CYTED, and grants CN2012/217 (REGICC), CN2011/034 ("Programa de consolidación y estructuración de unidades de investigación competitivas") and CN2012/211 ("Agrupación estratégica") from the Xunta de Galicia. Work also co-funded by FEDER (European Union).

References

1. Bemers-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American 284, 34–43 (2001)
2. Staab, S., Studer, R.: Handbook on ontologies. Springer (2009)
3. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220 (1993)
4. Gomez-Perez, A., Fernández-López, M., Corcho, O.: Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer (2004)
5. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges (2012)
6. Martínez-Romero, M., Vázquez-Naya, J.M., Pereira, J., Ezquerra, N.: Ontology alignment techniques. Encyclopedia of Artificial Intelligence 3, 1290–1295 (2008)
7. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. The Knowledge Engineering Review 18, 1–31 (2003)
8. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics IV, 146–171 (2005)
9. Qazvinian, V., Abolhassani, H., Haeri, S.H., Hariri, B.B.: Evolutionary coincidence-based ontology mapping extraction. Expert Systems 25, 221–236 (2008)
10. Wang, J., Ding, Z., Jiang, C.: GAOM: genetic algorithm based ontology matching. In: IEEE Asia-Pacific Conference on Services Computing, APSCC 2006, pp. 617–620. IEEE (2006)
11. Rijsbergen, C.J.: Information Retrieval. Butterworth (1979, 1997)