# GAOM: Genetic Algorithm based Ontology Matching

Junli Wang[1], Zhijun Ding[1, 2], Changjun Jiang[1]
*1. Department of Computer Science & Engineering,*
*Tongji University, Shanghai 201804, China*
*2. College of Information Science & Engineering,*
*Shandong University of Science & Technology, Qingdao 266510, China*
*wangjunli_1029@163.com*

## Abstract

*In this paper a genetic algorithm-based optimization procedure for ontology matching problem is presented as a feature-matching process. First, from a global view, we model the problem of ontology matching as an optimization problem of a mapping between two compared ontologies, and every ontology has its associated feature sets. Second, as a powerful heuristic search strategy, genetic algorithm is employed for the ontology matching problem. Given a certain mapping as optimizing object for GA, fitness function is defined as a global similarity measure function between two ontologies based on feature sets. Finally, a set of experiments are conducted to analysis and evaluate the performance of GA in solving ontology matching problem.*

## 1. Introduction

Nowadays, the Semantic Web community agrees on the fact that having a single domain ontology shared by a number of different applications may be not so feasible, since domain knowledge does strongly depend on the particular task at hand. Hence more and more ontologies with different terms or different taxonomies are being developed and many of them describe similar domain. For this reason, a key challenge is enabling the interoperability among different ontologies. In fact ontologies can interoperate only if correspondences between their elements have been identified and established[1]. As such, ontology engineers face the problem of how to map various different ontologies to enable a common understanding in order to support communication among existing and new domains[2].

Ontology matching is the process of determining a correspondence or mapping between two ontologies, that is, for every concept in an ontology finding out its corresponding concept in another ontology with the same or the closest intended meaning. Because ontologies can be compared from many different perspectives, numerous techniques for ontology matching have been proposed, some of them are based on semantic similarity function between two entities from two ontologies[3]. As an example, Rodriguez *et al*[3] considered three independent similarity assessments with respect to synonym sets of concepts, properties of concepts, and relations among concepts, and summed them with their weights to be a similarity function for two concepts. However, comparing relations among two concepts becomes a comparison between their related concept sets, which does not consider the types of relations, and then the structural characteristic of ontology can not be well employed.

Unlike the traditional approach comparing two concepts, a genetic algorithm (GA)[4] based optimization approach for the ontology matching problem is proposed in this paper. The reminder of this paper is organized as follows. In Section2, a series of definitions are presented for ontology model, and a mapping function is defined for ontology matching. Section3 demonstrates how GA could be employed for ontology matching problem. Experimental results in Section4 show that GA-based ontology matching algorithm has a satisfying performance. Section5 concludes our paper.

## 2. Ontology Mapping Model

### 2.1 Ontology model

Generally, every concept in an ontology has its explicit definition which is sufficiently detailed to capture the semantics of the domain. In our paper, four kinds of elementary factors are used to distinguish a concept within an ontology and they are concept name ($N$), property ($P$), instance ($I$), and relation ($R$). Since

IEEE
COMPUTER
SOCIETY

*N, P, I* are features related only with the concept, and *R* are features related the concept with another one, these four kinds of features can be classified as two kinds: intension for local information and extension for global information.

**Definition 1:** Intension of a concept $c$ is defined as a tuple $InT_c := (n_c, P_c, I_c)$, which describes the essence features of the concept, where

- $n_c$ is a name of the concept $c$. Every concept has only one distinctive name.

- $P_c$ is a set of properties related with the concept.

- $I_c$ is a set of instances associated with the concept.

**Definition 2:** Extension of a concept $c$ is defined as $ExT_c := R_c$, which profiles the structural property of the concept by its relations with other concepts, i.e. determines situation of the concept in the whole ontology, where

- $R_c$ is a set of relations that related the concept with other concepts in the ontology.

**Definition 3:** An ontology with $k$ concepts is modeled as a tuple $OM := (V_C, OntoInT_{V_C}, OntoExT_{V_C})$, where

$C = \{c_i \mid 1 \le i \le k\}$ is concept set of the ontology, and $c_i$ is one concept in the ontology, such that:

- $V_C = \{v_{c_i} \mid v_{c_i} = i, 1 \le i \le k\}$ is a set of sequence number of concept. where $v_{c_i}$ denotes that the $i$th concept in the ontology is ranked with a sequence number of $i$;

- $OntoInT_{V_C} = \{< v_{c_i}, Type, x_{c_i} >\}$ is a set of intensional features of the ontology, where $1 \le i \le k$, $Type = \{'N', 'P', 'I'\}$ and

$< v_{c_i}, Type, x_{c_i} > =$

$$
\begin{cases}
< v_{c_i}, 'N', n_{c_i} > & \text{if } Type='N' \\
< v_{c_i}, 'P', p_{c_i} > \quad p_{c_i} \in P_{c_i} & \text{if } Type='P' \\
< v_{c_i}, 'I', i_{c_i} > \quad i_{c_i} \in I_{c_i} & \text{if } Type='I'
\end{cases} ;
$$

- $OntoExT_{V_C} = \{< v_{c_i}, v_{c_j}, Type, r_{ij} >\}$ is a set of extensional features of the ontology, where $1 \le i \le k$, $1 \le j \le k$, $Type = 'R'$, $r_{ij} \in R_{c_i}$ and $< v_{c_i}, v_{c_j}, 'R', r_{ij} >$ represents that there exists a relationship $r_{ij}$ from concept $c_i$ to concept $c_j$.

**Definition 4:** An overall feature set of an ontology $F_O$ is defined as a combination of the intensional features set $OntoInT_{V_C}$ and the extensional features set $OntoExT_{V_C}$, that is, $F_O = OntoInT_{V_C} \cup OntoExT_{V_C}$.

**Definition 5:** Mapping function $M : V_C \to V_C'$ is defined as a mapping from one ontology $O_1$ to another ontology $O_2$, in which $V_C$ is the sequence number set in $O_1$, and $V_C'$ is the sequence number set in $O_2$.

As defined in Definition5, a mapping is a correspondence relationship between the taxonomies of two given ontologies. It states that any of concepts in $O_1$ should have a corresponding concept in $O_2$, and two different concepts in $O_1$ may correspond to the same concept in $O_2$.

## 3. Genetic algorithm for ontology matching

As a powerful search strategy based on natural selection and population genetics, genetic algorithm[4] outperforms conventional optimization methods such as the gradient ascent and simulated annealing.

The implementation of GA in application of the ontology matching problem incorporates three basic steps so that the algorithm is formulated for the specific application: the presentation of individual, i.e. the encoding mechanism of problem, and the formulation of the fitness function that gives to each individual a measure of performance.

### 3.1 Representation of Solution Strings

Let $n_1$ be the number of concepts in ontology $O_1$, and $n_2$ be the number of concepts in ontology $O_2$. GA is applied to find the optimal mapping function between concepts from these two ontologies, that is the solution string is corresponds to a mapping function $M$ defined in Definition6, indicating a mapping from each concept in $O_1$ to a corresponding concept in $O_2$. Each solution string or individual in the population would be a one-dimensional array with $n_1$ integer elements that may take values between 1 and $n_2$, denoted as: $N_1 N_2 \cdots N_{n_1}$, where $N_i = M(i) \in \{1, \cdots, n_2\}$, $i \in \{1, \cdots, n_1\}$ and this means that the $i$th concept in $O_1$ is mapped to the $N_i$th concept in $O_2$.

### 3.2 Fitness function

In order to determine the quality or performance of each encoded solution string in the population, the GA associates a fitness measure with each solution string. Here we define a global similarity measure function between two compared ontologies with respect to the mapping as fitness function in GA.

Since an ontology is represented as a collection of features, similarity between two compared ontologies will be described as a feature-matching process. Based

on the Tversky's similarity model and the set theory[5], a global similarity measure function between two ontologies $O_1$ and $O_2$ is defined as:

$$Similarity_{O_1,O_2}(M) =$$

$$\frac{f((F_{O_1} \cap F_{O_2})|M)}{f((F_{O_1} \cap F_{O_2})|M) + \alpha f((F_{O_1} - F_{O_2})|M) + \beta f((F_{O_2} - F_{O_1})|M)} \quad (1)$$

where $M$ is an individual of GA which corresponds to a mapping;

$F_{O_1}$ and $F_{O_2}$ are feature sets of ontology $O_1$ and $O_2$ respectively;

$(F_{O_1} \cap F_{O_2})|M$ stands for a set of matched elements of $F_{O_1}$ and $F_{O_2}$ with respect to the mapping $M$;

$(F_{O_1} - F_{O_2})|M$ and $(F_{O_2} - F_{O_1})|M$ are two sets of unmatched elements with respect to the mapping $M$;

$\alpha$, $\beta$ are two parameters between 0 and 1, which determine the relative importance of the two unmatched feature sets;

$f$ is a function defined as the cardinality of set.

Given two ontologies and a mapping $M$, when computing its fitness with the global similarity function in Equation (1), the problem is converted into how to compute $(F_{O_1} \cap F_{O_2})|M$, that is with respect to the mapping, identify how many elements between $F_{O_1}$ and $F_{O_2}$ will be matched. Because a feature set is described in two aspects: intensional features and extensional features, two kinds of match rules are defined as follows for deciding whether two elements respectively from $F_{O_1}$ and $F_{O_2}$ are matched.

**Intensional Rule:** for each intensional feature in $O_1$ $<v_c, Type_1, x_c> \in F_{O_1}$, if there exists $<v_{c'}, Type_2, x_{c'}> \in F_{O_2}$ where $v_{c'} = M(v_c)$, such that $Type_1 = Type_2$ and $x_c$ and $x_{c'}$ is local lexically matched, then $<v_c, Type_1, x_c>$ and $<v_{c'}, Type_2, x_{c'}>$ are matched intensionally.

**Extensional Rule:** for each extensional feature in $O_1$ $<v_c, v_{c_1}, 'R', r_c> \in F_{O_1}$, if there exists $<v_{c'}, v_{c_1'}, 'R', r_{c'}> \in F_{O_2}$, where $v_{c'} = M(v_c)$, $v_{c_1'} = M(v_{c_1})$, such that $r_c$ and $r_{c'}$ are local lexically matched, then $<v_c, v_{c_1}, 'R', r_c>$ and $<v_{c'}, v_{c_1'}, 'R', r_{c'}>$ are matched extensionally.

### 3.3 Genetic algorithm based ontology matching approach (GAOM)

When specific representation scheme, and a fitness function to evaluate the solutions have been decided, the final genetic algorithm for ontology matching can be developed. And the proposed genetic algorithm based ontology matching approach is given in Table 2, where elitist strategy[11] is employed to ensure that the most highly fit individuals in the population are passed on to the next generation without being altered by genetic operators.

**Table 1.** Genetic algorithm based ontology matching approach (*GAOM*)

| |
| --- |
| Input: feature sets $F_{O_1}$, $F_{O_2}$ of two compared ontologies $O_1$ and $O_2$ respectively, parameters of GA (size of population *pop_size*, crossover rate $p_c$, mutation rate $p_m$, and maximal generation *MaxGen*), $n_1$ and $n_2$ are the number of concepts in $O_1$ and $O_2$ respectively. |
| Output: the best mapping optimized by GA between the compared ontologies. |
| { Generate randomly an initial population of *pop_size* individuals /\*As described in Sect.3.1, every individual is an one-dimensional integer array with $n_1$ elements taking value from 1 to $n_2$, which corresponds to a mapping between concepts from $O_1$ to $O_2$. \*/ <br><br> while (certain termination criterion is not met) do /\* such as the number of generations is less than a given value of *MaxGen*, or fitness of the best individual is not close to the optimal *1\*/ <br> { <br> Cross over individuals according to a crossover probability $p_c$ to form new individuals; <br> Mutate individuals formed after crossover operation with a mutation probability $p_m$; <br> Evaluate every individual and compute its fitness value according to Equation (1); <br> According to their fitness and with a selection probability $p_s$, some individuals are selected to generate next new population. <br> The best individual in current population will be reserved into the next population;/\*Elitist strategy is employed\*/ <br> } <br> } |

## 4. Experiments

We conduct the experiments on OAEI 2005 benchmark test suit[1], and use three standard evaluation measures[6] to assess the results of our results, and they are respectively Precosion (Pre), Recall (Rec) and F-Measure(F1).

In this experiment, Elitist strategy is employed to save the current best solution after selection. And all related parameters in GA are assigned as follows: size of population is 80; crossover probability and mutation probability are respectively 0.9 and 0.001; and the max iteration count is 400.

With the fitness function in Equation (1) where $\alpha=0.8$, $\beta=0.2$, the experiments about matching performance are carried out by testing matching based on the mentioned benchmarks. And in Figure 1 and Table 2 some experimental results of several algorithms are given, in which falcon, dublin20 and foam are the three best matching algorithms mentioned in Ref.[7].

**Table 2.** Performance comparison

|     |     | Falcon | Dublin20 | Foam | GAOM |
|-----|-----|--------|----------|------|------|
| *Pre* | 1xx | 1.00 | 1.00 | 0.98 | 1.00 |
|     | 2xx | 0.90 | 0.94 | 0.89 | 0.92 |
|     | 3xx | 0.93 | 0.67 | 0.92 | 0.89 |
| *Rec* | 1xx | 1.00 | 0.99 | 0.65 | 1.00 |
|     | 2xx | 0.89 | 0.71 | 0.69 | 0.80 |
|     | 3xx | 0.83 | 0.60 | 0.69 | 0.82 |
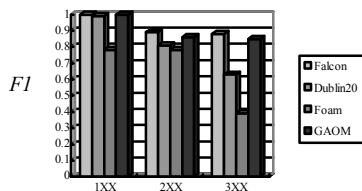


**Figure 1.** *F1* of four compared algorithms

As shown in Table 2 and Figure 1, our algorithm GAOM is better mostly than Dublin20 and Foam, next to Falcon slightly. The reason is that Falcon is an integrated system which could take external mapping from other algorithm as inputs. The other two algorithms coincide with the generic alignment process proposed by Ehrig and Staab in Ref.[8], they are all emphasize the similarity between two concepts locally. However our algorithm defines global similarity function under a mapping, which can well employed the essence and structural properties of ontology, so it has a better performance.

## 5. Conclusion

In this paper, we have presented a genetic algorithm based approach for solving the ontology matching problem. For the purpose of better and precise representation of ontology feature, we defined features of ontology from 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 genetic algorithm is used to achieve an approximate optimal solution. Finally, we conducted a set of experiments, which analyze and evaluate the performance of GA in solving ontology matching problem, and the experimental results shown that our ontology matching algorithm has good performance.

## 6. Acknowledgement

## 7. References

[1] Kalfoglou Y, Schorlemmer M., "Ontology mapping: the state of the art", The Knowledge Engineering Review, Vol.18, 2003, pp.1-31.
[2]Ding, Y., & Foo, S., "Ontology research and development, Part 2 - A review of ontology mapping and evolving", Journal of Information Science, Vol.28, No.5, 2002, pp.375-388.
[3] Rodriguez M A, Egenhofer M J., "Determining semantic similarity among entity classes from different ontologies", IEEE Transactions on Knowledge and Data Engineering, Vol. 15, 2003, pp.442-456.
[4] DeJong, K. A., "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", Ph.D. thesis. University of Michigan Press, Ann Arbor, 1975.
[5] Tversky, "A. Features of similarity. Psychological Review", Vol. 84, 1977, pp.327–352.
[6]Van Rijsbergen C J., "Information Retrieval", Butterworths, London, 1979.
[7]Euzenat J, Stuckenschmidt H, Yatskevich M., "Introduction to the Ontology Alignment Evaluation 2005", In Proc. K-Cap 2005 workshop on Integrating ontology, Banff, 2005, pp.61-71.
[8]Ehrig, M. and Stab, S., "QOM - Quick Ontology Mapping", In Proceedings of the third International Conference on Semantic Web, LNCS 3298, Springer, Hiroshima, Japan, 2004, pp.683-696.

---

1 http://oaei.inrialpes.fr/2005/benchmarks/