



Optimizing ontology alignments through a Memetic Algorithm using both MatchFmeasure and Unanimous Improvement Ratio



Xingsi Xue^{a,b}, Yuping Wang^{a,*}

^a School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, China

^b School of Information Science and Engineering, Fujian University of Technology, Fuzhou, Fujian, China

ARTICLE INFO

Article history:

Received 1 April 2014

Received in revised form 14 February 2015

Accepted 1 March 2015

Available online 5 March 2015

Keywords:

Ontology alignment

Memetic Algorithm

MatchFmeasure

Unanimous Improvement Ratio

ABSTRACT

There are three main drawbacks of current evolutionary approaches for determining the weights of ontology matching system. The first drawback is that it is difficult to simultaneously deal with several pairs of ontologies, i.e. finding a universal weight configuration that can be used for different ontology pairs without adjustment. The second one is that a reference alignment between two ontologies to be aligned should be given in advance which could be very expensive to obtain especially when the scale of ontologies is considerably large. The last one arises from f-measure, a generally used evaluation metric of the alignment's quality, which may cause the bias improvement of the solution. To overcome these three defects, in this paper, we propose to use both MatchFmeasure, a rough evaluation metric on no reference alignment to approximate f-measure, and Unanimous Improvement Ratio (UIR), a measure that complements MatchFmeasure, in the process of optimizing the ontology alignments by Memetic Algorithm (MA). The experimental results have shown that the MA using both MatchFmeasure and UIR is effective to simultaneously align multiple pairs of ontologies and avoid the bias improvement caused by MatchFmeasure. Moreover, the comparison with state-of-the-art ontology matching systems further indicates the effectiveness of the proposed method.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Ontologies are regarded as the solution to data heterogeneity on the semantic web. However, because of human subjectivity, the ontologies could themselves introduce heterogeneity: given two ontologies, one entity can be given different names or simply be defined in different ways. Addressing this heterogeneity problem requires to identify correspondences between entities of various ontologies. This process is commonly known as ontology alignment which can be described as follows: given two ontologies with each describing a set of discrete entities (which can be classes, properties, instances, etc.), we have to find the relationships (e.g., equivalence or subsumption) that hold between these entities [1].

It is highly impractical to align the ontologies manually when the size of ontologies is considerably large. Thus, numerous ontology matching systems have arisen over the years. Each of them could provide, in a fully automatic or semi-automatic way, a numerical value of similarity between elements from separate ontologies that can be used to decide whether those

* Corresponding author.

E-mail address: ywang@xidian.edu.cn (Y. Wang).

elements are semantically similar or not. Since none of the similarity measures could provide the satisfactory result independently, most ontology matching systems combine a set of different similarity measures together by aggregating their aligning results. How to select the appropriate similarity measures, weights and thresholds in ontology aligning process in order to obtain a satisfactory alignment is called meta-matching which can be viewed as an optimization problem and be addressed by evolutionary approaches like Memetic Algorithms (MA).

Since modeling the meta-matching problem is a complex (nonlinear problem with many local optimal solutions) and time-consuming task (large scale problem), particularly when the number of similarity measures is significantly large, approximate methods are usually used for computing the parameters. From this point of view, evolutionary optimization methods could represent an efficient approach for addressing this problem. However, the slow convergence and premature convergence are two main shortcomings of the classical evolutionary algorithms (e.g. Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithm) for this kind of problem. It makes these algorithms incapable of effectively searching the optimal solution for large scale and complex problems. Starting from these considerations, our work investigates the methodology of using an emergent class of evolutionary algorithms, named Memetic Algorithms (MA), to efficiently tackle the meta-matching problem. MA is a population-based search method which combines genetic algorithms (global search) and local refinements (local search). This marriage between global search and local search allows keeping high population diversity via strong mutation (thus, reducing the possibility of the premature convergence) and increasing the convergence speed via the local search (in fact, local search can greatly improve the solution quality and thus make the solution approaches to optimal solution more quickly). Therefore, MA is very suitable to the problem considered.

Nevertheless, there are three main drawbacks of current evolutionary approaches for determining the weights of ontology matching systems. The first drawback is that it is difficult to simultaneously deal with several pairs of ontologies, i.e. finding a universal weight configuration that can be used for different ontology pairs without adjustment. The second one is that a reference alignment between two ontologies to be aligned should be given in advance which could be very expensive to obtain especially when the scale of ontologies is considerably large. The last one arises from f-measure, a generally used evaluation metric of the ontology alignment's quality, which may cause the bias improvement of the solution. To be specific, the improvement of f-measure does not say anything about whether both evaluation metrics involved (i.e. recall and precision) are simultaneously improved or not. In other words, no matter how large a measured improvement in f-measure is, it can still be extremely dependent on how we are weighting the evaluation metrics involved. To overcome these three defects, in this paper, we propose to use both MatchFmeasure, a rough evaluation metric on no reference alignment to approximate f-measure, and Unanimous Improvement Ratio (UIR) [27], a measure that complements MatchFmeasure, in the process of optimizing the ontology alignments by Memetic Algorithm (MA). In particular, our proposed method applies to the specific scenario that the target ontologies are the variations of the same source ontology. For example, given a source ontology O_A , three target ontologies O_B , O_C and O_D which are different variations of O_A in terms of different lexical, linguistic and ontology structure respectively, the goal of our proposed method is to determine the optimal parameters, in terms of both MatchFmeasure and UIR, for the ontology matching tasks that match O_A with O_B , O_A with O_C and O_A with O_D respectively. Moreover, the obtained parameter set could be reused in the task of matching O_A with O_E which is another target ontology that has different lexical, linguistic and ontology structure from O_A at the same time.

The remainder of the paper is organized as follows: Section 2 gives a brief foundation of our work; Section 3 presents the related work about MA and evolutionary algorithm for the ontology alignment problem; Section 4 provides a detailed description of the basic concepts of the similarity measures, the aggregation strategy and the ontology alignment evaluation metrics; Section 5 presents the details of MA based on MatchFmeasure and UIR; Section 6 shows the experimental results of our approach; finally, in Section 7, we draw conclusions and propose the future improvement.

2. Foundation

There are numerous definitions of ontology over years. But the most frequently referenced one was given by Gruber in 1993 which defined the ontology as an explicit specification of a conceptualization. For the convenience of understanding the work in this paper, the ontology is defined as following:

Definition 1. (See [25].) An ontology is a 9-tuple $O = (C, P, I, A, \leq_C, \leq_P, \phi_{CP}, \phi_{CI}, \phi_{PI})$, where:

- C is a nonempty set of classes,
- P is a nonempty set of properties,
- I is a set of instances (it can be empty),
- A is a set of axioms which should not be empty,
- \leq_C is a partial order on C , called class hierarchy or taxonomy,
- \leq_P is a partial order on P , called property hierarchy,
- $\phi_{CP} : P \rightarrow C \times C$ is a function which associates a property $p \in P$ with two linked classes through the property p . We denote the domain by $dom(p) := \pi_1(\phi_{CP}(p))$ and the range by $ran(p) := \pi_2(\phi_{CP}(p))$ where $\pi_1()$ and $\pi_2()$ are two functions obtaining the domain class and range class respectively,
- $\phi_{CI} : C \rightarrow \mathcal{P}(I)$ is a function which associates a concept $c \in C$ with a subset of I representing the instances of the concept c ,

- $\phi_{PI} : P \rightarrow \mathcal{P}(I^2)$ is a function which associates a property $p \in P$ with a subset of Cartesian product $I \times I$ representing the pair of instances related through the property p .

In general, classes, properties and individuals are referred as entities.

At present, ontologies are viewed as a practical way to conceptualize information that is expressed in electronic format, and are used in many applications from different areas. However, certain systems that encompass a large number of components associated with different domains would generally require the use of different ontologies. In such cases, using ontologies would not reduce heterogeneity but rather would recast the heterogeneity problem into a different (and higher) framework wherein the problem becomes one of ontology alignment, thereby allowing the more efficient exchange of information and knowledge derived from different (heterogeneous) data bases, knowledge bases, and the knowledge contained in the ontologies themselves.

The tools, designed to automatically identify the correspondences that may exist between entities of different ontologies, are called ontology alignment systems. In particular, an ontology alignment system executes an ontology alignment process which can be defined as follows:

Definition 2. (See [26].) An alignment between two ontologies is a set of mapping elements. A mapping element is a 5-tuple (id, e, e', n, R) , where:

- id is a unique identifier for the mapping,
- e and e' are the entities of the first and the second ontologies, respectively,
- n is a confidence measure in some mathematical structure (typically in the range $[0, 1]$) holding for the correspondence between entities e and e' ,
- R is a relation, e.g. equivalence, more general and disjointedness, of the correspondence between entities e and e' .

In principle, all relations between entities in the given ontology language can be used as the correspondence relation, and the interpretation of correspondences and alignments is strongly case-dependent. However, in many cases, a correspondence between ontological entities is always thought of expressing the “equivalent” or at least somewhat “similar” entities. A common assumption is to regard a correspondence as equivalence axiom for two corresponding entities. Furthermore, the ontology alignment process can be defined as follows:

Definition 3. (See [26].) The alignment process can be seen as a function Φ where given a pair of ontologies O and O' , a partial (and optional) input alignment A , a set of parameters p , a set of resources r , returns a new alignment A' :

$$A' = \Phi(O, O', A, p, r).$$

The key issue in ontology alignment process is finding which entity in one ontology corresponds (in terms of meaning) to another entity in another ontology. Essentially, one might say that ontology alignment can be reduced to define a similarity measure between entities in different ontologies and select a set of correspondences between entities of different ontologies with the highest similarity measures [7]. Nevertheless, since there does not exist a similarity measure works better than all the other ones in every scenario, an alignment system typically calculates the similarity between a pair of entities from two different ontologies by means of a set of multiple similarity techniques, and the obtained similarity values are combined in a unique similarity measure. However, how to find the adequate values for the weights to aggregate various similarity values is not an easy task. It is considered to be essential to obtain high quality alignments in an automatic manner, and therefore, to reach an adequate semantic interoperability between systems in domains.

The concept of the MA was first introduced by Moscato and Norman [3] in 1992 to present an evolutionary algorithm in which local search is used. This idea has been further formalized by Radcliffe & Surrey [4] and a comparison between MA and GA is made. MA may be considered as a synergy between population-based approaches (in generally, a population consists of multiple individuals or solutions) and separate local improvement methods inspired by Darwinian's principles of natural evolution and Dawkins' notion of a meme, defined as a unit of cultural evolution that is capable of local refinements [5]. The main advantage of MA is that the space of the possible solutions is reduced to the subspace of local optima, which ensures MA not only converge to high-quality solutions, but also search vast solution spaces more efficiently than conventional evolutionary algorithms. Thus, MA has been applied to many different application domains that range from scheduling and floor-planning problems, to pattern recognition, vehicle routing, control systems, aircraft, and drug design, and so forth [28].

Aiming at providing the best value according to the proposed alignment evaluation metric, this work proposes an MA-based approach to automatically find the best manner of aggregating different similarity measures into a single similarity metric and determine the threshold for filtering the aggregated alignment.

3. Related work

Lately, the focus of fully automatic or semi-automatic matching systems is on meta-matching. Meta-matching does not use parameters from an expert, but selects the parameters according to a training benchmark, which is a set of ontologies that have been previously aligned by an expert. One group of the meta-matching techniques is called heuristic meta-matching, where the most outstanding approaches are based on evolutionary algorithms.

Among those meta-matching systems using evolutionary algorithm, the most notable system is GOAL (Genetics for Ontology Alignments) [6]. Although GOAL does not directly compute the alignment between two ontologies, it determines, through a GA, the optimal weight configuration for a weighted average aggregation of several similarity measures by considering a reference alignment. The same idea of implementing a meta-matching system to combine multiple similarity measures into a single aggregated metric is also developed in papers [7,8]. In paper [24], given a Partial Reference Alignment (PRA), i.e. a part of golden alignment given by the domain expert, and on the basis of the PRA based quality measures, various approaches including GA are utilized to tune the parameters of the meta-matching system. Inspired by GA, Vitiello et al. employ MA in the alignment problem [30]. Since the MA has the capability of realizing local search process within the successive generations, it improves the performance of genetic approach in both quality of solutions and computational efficiency. However, all the aforementioned approaches are not able to determine a universal weight configuration that can be used for different ontology pairs without adjustment, and the quality measures used by them more or less require the involvement of the domain experts, and the shortcomings of f-measure are not considered in the process of tuning the meta-matching system. To this end, we propose to use MA with both MatchFmeasure and UIR in our work to tackle these issues.

Moreover, ontology alignment system MapPSO exploits a multi-objective evaluation [20]. However, precisely, this work does not address the meta matching problem, but it directly solves an optimization problem by using an evaluation method based on multiple objectives following an “a priori” approach for the ontology alignment problem, i.e. they aggregate all objectives in a weighted function. In order to overcome the well-known drawbacks of “a priori” methods, [21] proposes to apply NSGA-II to ontology alignment problem, and the results show the significant improvement of semantic inter-operability by finding high quality solutions that cannot be detected by “a priori” approaches. Comparing with these approaches which try to get a Pareto front resulting from optimizing objectives such as precision and recall in parallel, MA has the advantage of lower time and memory consumption and is also able to obtain high quality alignment.

4. Preliminaries

4.1. Similarity measures

Typically, the most commonly used similarity measures are syntactic, linguistic and taxonomy-based measures. In particular, according to [22], syntactic measure belongs to linguistic measure in schema-only based category, linguistic measure is a kind of constraint-based measure in instance/contents-based category, and taxonomy-based measure belongs to constraint-based measure in schema-only based category. In the following, we present some common similarity measures belonging to these three categories.

4.1.1. Syntactic measures

Syntactic measures compute a string distance between named entity identifiers. In the considered ontologies, each entity is identified by a URI and can optionally have a natural language label [20]. There are a number of string distance methods such as Levenstein distance [9], Jaro–Winkler distance [10], SMOA distance [15], and so on. However, according to [15], the SMOA distance is shown to be a very-well performed measure for the ontology alignment problem. Therefore, in this work, we choose the SMOA distance as the string distance measure. Formally, given two strings s_1 and s_2 , the SMOA distance between them can be defined by the following equation:

$$SMOA(s_1, s_2) = Comm(s_1, s_2) - Diff(s_1, s_2) + WinklerImpr(s_1, s_2) \quad (1)$$

where $Comm(s_1, s_2)$ stands for the commonality between s_1 and s_2 , $Diff(s_1, s_2)$ for the difference and $WinklerImpr(s_1, s_2)$ for the improvement of the result using the method introduced by Winkler in [10].

4.1.2. Linguistic measures

Linguistic based matchers compute a similarity between entity names/labels, based on synonymy, hypernymy, and other linguistic relations [20]. To compute the linguistic similarity or inversely the linguistic distance, a lexicon and a thesauri are needed, and the most popular one being used to identify the relationships between entities is WordNet [11] which is an electronic lexical database where various senses of words are put together into sets of synonyms. Given two words w_1 and w_2 , $LinguisticDistance(w_1, w_2)$ equals:

- 1, if words w_1 and w_2 are synonymous,
- 0.5, if word w_1 is the hypernym of word w_2 or vice versa,
- 0, otherwise.

4.1.3. Taxonomy-based measures

Taxonomy-based measures compute a similarity between ontological entities based on their structural properties according to the “ontology graph” [20]. In our work, the taxonomy-based measure is computed based on the intuition that elements of two distinct ontologies are similar when their adjacent elements are similar. In particular, the taxonomy-distance is calculated through the well known Similarity Flooding (SF) algorithm [23] where an iterative fix-point computation (see also Eq. (2)) is utilized to produce an alignment between the elements of two ontologies.

$$\delta^{i+1} = \text{normalize}(\delta^i + f(\delta^i)) \quad (2)$$

where function f increments the similarity value of an element pair based on the similarity of its neighbors, and the previous iteration's value (δ^i) changes in each variation. About the details of the SF algorithm, see also [23].

4.1.4. Aggregation strategy

Since the application of a single similarity measure is often not enough to produce an acceptable output alignment, the common strategy is to combine different similarity measures to compute a unique confidence value as an aggregated similarity value. Obviously, the quality of alignment is strongly dependent on the selection of the appropriate aggregation strategy. However, the determination of an adequate aggregation strategy is not an easy task, which is a complex process and is normally achieved manually by an expert or by means of general approaches (e.g. maximum, minimum, geometric mean, harmonic mean function, etc.). However, these approaches are not appropriate to provide optimal results for all alignment problems. Therefore, in this work, we choose the weighted average strategy to aggregate different similarity measures into a single similarity metric, and further utilize MA to automatically find the best manner of aggregating different similarity measures into a single similarity metric. In general, the weighted average aggregation is defined as follows:

$$\phi(\vec{s}(c), \vec{w}) = \sum_{i=1}^{num} w_i s_i(c) \quad (3)$$

where:

- $\sum_{i=1}^{num} w_i = 1$ and $w_i \in [0, 1]$,
- $\vec{s}(c)$ is the vector of similarity measure results,
- \vec{w} is the vector of weights,
- num is the number of similarity measures.

Since the quality of resulting alignment, the correctness and completeness of the correspondences found already need to be assessed, we will introduce some conformance measures which are derived from the information retrieval field in the next section [12].

4.2. MatchFmeasure and UIR

4.2.1. Alignment evaluation metrics on reference alignment

The alignment is normally assessed on the basis of two measures commonly known as recall and precision. The recall (or completeness) measures the fraction of the number of the correct mapping pairs found over the total number of the existing correct ones. Typically, the recall is balanced against precision (or correctness), which measures the fraction of the number of the correct mapping pairs found over the number of the found ones. Given a reference alignment R and some alignment A , recall and precision are given by the following formulas:

$$\text{recall} = \frac{|R \cap A|}{|R|} \quad (4)$$

$$\text{precision} = \frac{|R \cap A|}{|A|} \quad (5)$$

In most instances, it requires considering both recall and precision to compare alignments' performance. The most common combining function is the f-measure which is defined as follows:

$$f_{\text{measure}} = \frac{\text{recall} \cdot \text{precision}}{\alpha \cdot \text{recall} + (1 - \alpha) \cdot \text{precision}} \quad (6)$$

where α is the relative weight of recall and precision which is in the range $[0, 1]$. When $\alpha = 0$ or 1 , f-measure can be transformed into recall or precision; when $\alpha = 0.5$, both recall and precision have the same relative weight, and f-measure computes their harmonic mean.

4.2.2. Alignment evaluation metrics on no reference alignment

Although the alignment evaluation metrics, recall, precision and f-measure, can reflect the quality of the resulting alignment, they require that the perfect matching result, i.e. the reference alignment, should be given in advance. However, this perfect match result is generally unknown for difficult real-life match problems, especially for large heterogeneous ontologies. Therefore, supposing the golden alignment is 1:1 (i.e. the entity of one ontology can correspond to only one entity of the other) and complete (i.e. every entity of an ontology participates in a correspondence), in this paper, we use the rough alignment evaluation metrics, which do not rely on the reference alignment, to approximate the recall and precision.

Match coverage [16], the fraction of entities which exist in at least one correspondence in the resulting alignment in comparison to the total number of entities in the ontology, is used as a substitute for recall. Two formulas of match coverage are presented as follows:

$$MatchCoverage_{O_1} = \frac{|S_{O_1-Match}|}{|S_{O_1}|} \in [0, 1] \quad (7)$$

$$MatchCoverage_{O_2} = \frac{|S_{O_2-Match}|}{|S_{O_2}|} \in [0, 1] \quad (8)$$

where:

- S_{O_1} and S_{O_2} are the sets of all entities of ontology O_1 and O_2 respectively.
- $S_{O_1-Match}$ and $S_{O_2-Match}$ are the sets of matched entities of ontology O_1 and O_2 respectively.

In addition, we define the combined match coverage for a matching result as follows:

$$MatchCoverage = \frac{|S_{O_1-Match}| + |S_{O_2-Match}|}{|S_{O_1}| + |S_{O_2}|} \in [0, 1] \quad (9)$$

With respect to precision, we introduce match ratio [16], the ratio between the number of the found correspondences and the number of the matched entities. The intuition is that the precision of a match result is better if an entity is not loosely matched to many other concepts but only to fewer ones [16]. Therefore, the match ratio can be defined as follows:

$$MatchRatio_{O_1} = \frac{|Corr_{O_1-O_2}|}{|S_{O_1-Match}|} \in [1, \infty) \quad (10)$$

$$MatchRatio_{O_2} = \frac{|Corr_{O_1-O_2}|}{|S_{O_2-Match}|} \in [1, \infty) \quad (11)$$

where:

- $Corr_{O_1-O_2}$ is the set of correspondences in a resulting alignment,
- $S_{O_1-Match}$ and $S_{O_2-Match}$ are the sets of matched entities of ontology O_1 and O_2 respectively.

Analogously, the combined match ratio can be defined as follows:

$$MatchRatio = \frac{2 * |Corr_{O_1-O_2}|}{|S_{O_1-Match}| + |S_{O_2-Match}|} \in [1, \infty) \quad (12)$$

High match ratio indicates entities are mapped to many other entities. Match ratio that is close to 1.0 indicates the high precision. Therefore, for the convenience, we define Frequency below as the rough approximation for precision.

$$Frequency = \frac{1}{MatchRatio} \in (0, 1] \quad (13)$$

In our work, we use MatchFmeasure, which is defined as follows, to approximate the traditional f-measure.

$$MatchFmeasure = \frac{MatchCoverage \cdot Frequency}{\alpha \cdot MatchCoverage + (1 - \alpha) \cdot Frequency} \quad (14)$$

where α is the relative weight of MatchCoverage and Frequency and is in the range [0, 1].

4.2.3. UIR

One of the shortcomings of MatchFmeasure is that the improvement of it does not say anything about whether both MatchCoverage and Frequency are simultaneously improved or not. In other words, no matter how large a measured improvement in MatchFmeasure is, it can still be extremely dependent on how we are weighting the individual metrics in that measurement [2]. For example, if an individual (or a solution of MA which could further be used to generate an alignment) A improves another individual B in Frequency with a loss in MatchCoverage, MatchFmeasure may say that A is better

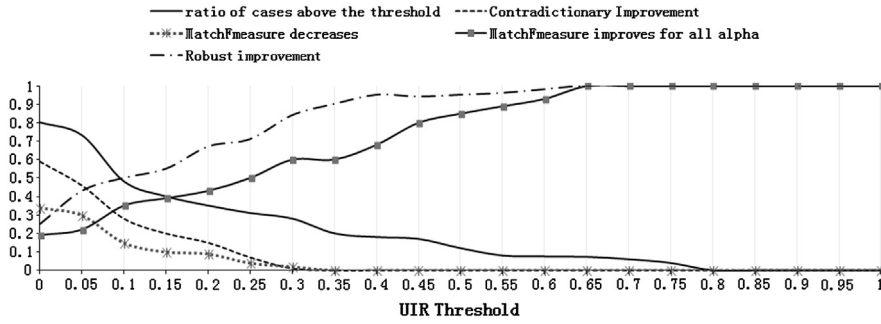


Fig. 1. Improvement detected across UIR thresholds.

than B , depending on the relative weight α of Frequency and MatchCoverage. Therefore, it is controversial to rank the quality of individuals produced by MA according to the MatchFmeasure only since an overall improvement in MatchFmeasure is often derived from an improvement in one of the metrics at the expense of a decrement in the other.

To overcome the shortcoming of MatchMeasure, we employ the UIR which is a measure that allows to compare two individuals using MatchCoverage and Frequency without dependency on the relative weight α in MatchFmeasure. In particular, UIR takes the results of multiple testing cases into consideration to judge whether the improvement of one individual on the other is robust on various α in MatchFmeasure. For two individuals A and B , UIR can be calculated by the following formula:

$$UIR(A, B) = \frac{(|T_A| - |T_B|)}{|T|} \quad (15)$$

where:

- T_A is the set of cases for which the Frequency and MatchCoverage achieved by individual A are greater than or equal to those achieved by individual B ,
- T_B is the set of cases for which the Frequency and MatchCoverage achieved by individual B are greater than or equal to those achieved by individual A ,
- T is the set of all cases.

In this work, we set $\alpha = 0.5$ to favor neither Frequency nor MatchCoverage. When $\alpha = 0.5$, we can use MatchFmeasure to identify an order relation for individuals by which we can determine whether one individual is better than another. In order to estimate whether this order relation is robust against different α values, we need to determine the UIR threshold in the context of our work so that UIR above the threshold guarantees that this order relation is robust, and at the same time the threshold is not too strong to be satisfied in practice. According to the approach proposed in [2], we take the weights being all 0.25 and the threshold being 0.80 as the baseline parameter set and randomly select 20 parameter sets, and use the datasets listed in Table 3 to carry out the experiments. The average results are shown in Fig. 1.

As shown in Fig. 1, a UIR threshold of 0.25 accepts around 30% of all system pairs. For this UIR threshold value, the number of contradictory improvements and the number of cases where MatchFmeasure decreases are low (7% and 4% respectively). Also, MatchFmeasure increases for all α values in 50% of the cases, and improvements are robust in 71% of the cases. Therefore the rule of thumb proposed in [2] is also reasonable in our work, i.e. if $UIR(A, B) > 0.25$, then an observed improvement of individual A over B in MatchFmeasure where $\alpha = 0.5$ is robust. In addition, the approach to rank the quality of an individual based on both MatchFmeasure and UIR is illuminated in Section 5.5.

The main advantage of UIR is that no metric weighting is necessary. However, there remain two main limitations of UIR. First, the unanimous improvement is not transitive [2]. Therefore, it is not possible to define a linear individual ranking based on UIR. In addition, there is some information lost when the systems are compared if the ranges in evaluation results are not considered. To tackle these limitations, we propose an MA utilizing both the MatchFmeasure and UIR to optimize multiple ontology alignments simultaneously, which will be discussed in detail in Section 5.

5. MA using MatchFmeasure and UIR

There are some preparatory steps before deploying the MA. First, the similarity measures are chosen. Second, given a set of pairwise ontologies as the input, the similarity values between their entities through various measures are calculated separately and all the results are stored in XML format in order to avoid recalculating the similarity during the process of running MA. For example, ontology O_1 and O_2 are taken as input, and syntactic measure, linguistic measure and taxonomy-based measure, are chosen as the similarity measures, in the second step, the similarity values between all entities of O_1 and O_2 are calculated using syntactic measure, linguistic measure and taxonomy-based measure respectively and the re-

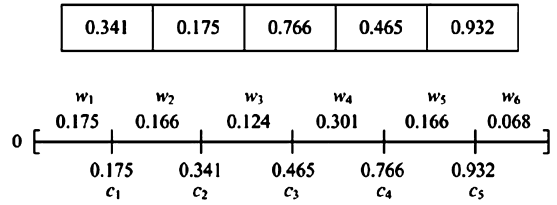
Table 1

The outline of Memetic Algorithm.

```

Initialize the Generation  $t = 0$ ;
Initialize the Population  $P_t$ ;
Evaluate( $P_t$ );
while  $t < \text{MaxGeneration}$  do
   $P'_t = \text{Crossover}(P_t)$ ;
   $P''_t = \text{Mutation}(P'_t)$ ;
  Evaluate( $P''_t$ );
  localSearch();
  saveElite();
   $P_t = \text{Select}(P''_t)$ ;
   $t = t + 1$ ;
end while

```

**Fig. 2.** An example shows the encoding and decoding mechanisms of the weights.

spective results are saved in separate files in XML format. Therefore, in our work, each pair of ontologies corresponds to three files in XML format, and what we do further is to utilize MA to determine three parameters for aggregating the results in these three files into one, and a threshold for filtering the aggregated result to determine the optimal alignment. The outline of MA in our work is presented in Table 1.

In the following, five basic steps of MA are presented.

5.1. Encoding mechanism

Note that we want to determine the weights (associated with the similarity measures) and the threshold (to decide whether a pair of entities is an alignment or not). Thus we use the weights and the threshold to form an individual, i.e. one individual consists of the two parts where one stands for several weights and the other for threshold. In particular, concerning the characteristics of the weights which are mentioned in formula (3). Our encoding mechanism for the weights is as follows. If p is the number of weights required, generate $(p - 1)$ cut points $c' = \{c'_1, c'_2, \dots, c'_{p-1}\}$ in $(0, 1)$ in some way, then queue them in ascending order and denote the queued cut points by $c = \{c_1, c_2, \dots, c_{p-1}\}$ (i.e. $0 \leq c_1 \leq c_2 \leq \dots \leq c_{p-1} \leq 1$). Now q weights can be represented by these cut points $c = \{c_1, c_2, \dots, c_{p-1}\}$. The decoding mechanism for weights is as follows:

$$w_k = \begin{cases} c_1, & k = 1 \\ c_k - c_{k-1}, & 1 < k < p \\ 1 - c_{p-1}, & k = p \end{cases} \quad (16)$$

Thus, an individual can be encoded (represented) by a p dimensional vector $(c_1, c_2, \dots, c_{p-1}, c_p)$ where c_p is a possible threshold value.

Fig. 2 presents an example that shows the encoding and decoding mechanisms of the weights. In this example, six weights are considered, which would measure the contribution of six different similarity measures to an aggregated metric. As can be observed, the chromosome values are ordered before calculating the set of weights.

5.2. Initialization

In conventional MA, the initial population would be constituted by a randomly generated group of individuals. While in our work, the Good-Lattice Point Method (GLPM) [13] is employed in the initialization process to accelerate the convergence of the algorithm. GLPM is a method of approximate uniform design which is presented as follows: let $a = \{(x_1, x_2, \dots, x_u) \mid 0 \leq x_1, x_2, \dots, x_u \leq 1\}$, $\langle \theta \rangle$ be the decimal part of θ , p_1, p_2, \dots, p_u be the first u prime numbers and $(\gamma_1, \gamma_2, \dots, \gamma_u) = (\sqrt{p_1}, \sqrt{p_2}, \dots, \sqrt{p_u})$, q be the number of uniform distributed points in a , then $\{(\langle k\gamma_1 \rangle, \langle k\gamma_2 \rangle, \dots, \langle k\gamma_u \rangle) \mid k = 1 \sim q\}$ are the uniform distributed points in a [13]. For instance, given the dimension of the problem is four, namely $c = \{(x_1, x_2, x_3, x_4) \mid 0 \leq x_1, x_2, x_3, x_4 \leq 1\}$, then the first four prime numbers are 2, 3, 5 and 7 respectively

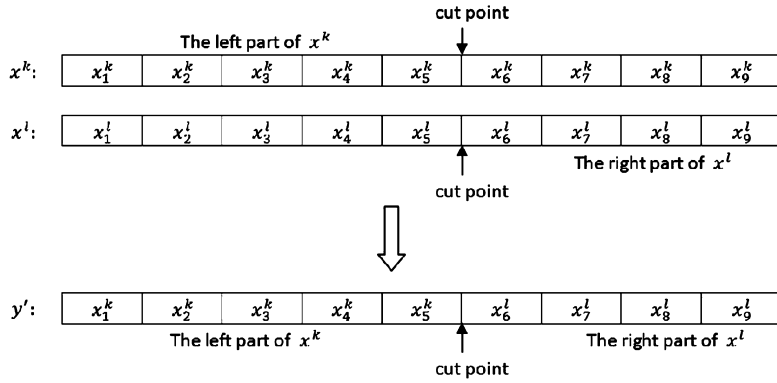


Fig. 3. An example shows the procedure of crossover.

and $(\gamma_1, \gamma_2, \gamma_3, \gamma_4) = (\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{7})$. Supposing three uniformly distributed points in c are desired, namely $q = 3$, these points are $(0.41, 0.73, 0.24, 0.65)$, $(0.83, 0.46, 0.47, 0.29)$ and $(0.24, 0.20, 0.71, 0.94)$.

In this work, assuming m is the population size, the number of individuals generated by GLPM is $\lfloor 4m/3 \rfloor$. Then, those individuals are queued in descending order according to their fitnesses. Finally, the first m individuals are chosen to form the initial population.

5.3. Fitness functions

Fitness function evaluates the quality of the alignments obtained by using the weights and the threshold encoded in the individual. In our work, the fitness function should take into account the quality of all the alignments. This is done by means of the average of all the MatchFmeasure as follows:

$$\text{fitness} = \frac{\text{MatchFeature}_1 + \dots + \text{MatchFeature}_s}{s} \quad (17)$$

where s is the number of pairs of ontologies to be aligned.

5.4. Genetic operators

5.4.1. Selection

Like in nature, since the most suitable individuals, i.e. those individuals with the highest fitness value, can potentially provide the best solutions to the problem, they should have more opportunities to be selected to make the reproduction. However, reproduction opportunities of the less suitable individuals should not be completely removed, because it is important to keep diversity in the population. In this article, we use a roulette wheel selection method where an individual is given a probability of being selected that is directly proportionate to its fitness value. So the most suitable individuals will have more opportunities of reproduction, while the less suitable individuals also have the chance of reproduction. Two individuals are then chosen randomly based on these probabilities to produce offsprings.

5.4.2. Crossover

For each pair of the selected individuals (called parents), the crossover operator generates two children, which are obtained by mixing the genes of the parents. Crossover is applied with a certain probability, a parameter of the GA. In this work, we use the widely used one-cut-point crossover operator. First, a cut position in two parents is randomly determined and this position is a cut point which cuts each parent into two parts: the left part and the right part. Then, the right parts of them are switched to form two children. An example of the crossover is presented in Fig. 3.

5.4.3. Mutation

Mutation operator assures diversity in the population and prevents premature convergence. In our work, for each component in the individual, we check if the mutation could be applied according to the mutation probability, and if it is, the value of the component is then randomly changed to a value in $(0, 1)$.

In Fig. 4, an example is presented to show the procedure of mutation, and the symbol y and y' stands for the original individual and the offspring generated by the mutation respectively.

5.5. Determine the best individual by elitist strategy

Elitist strategy puts the best individual (elite) of the current population unaltered in the next population. This ensures the survival of the elite that has been obtained up to the moment. In our work, we utilize MatchFmeasure and UIR to obtain

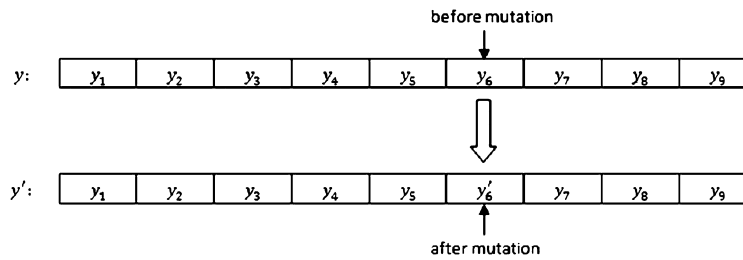


Fig. 4. An example shows the procedure of mutation.

Table 2

Brief description of benchmarks.

ID	Brief description
101–104	The ontologies under alignment are the same or the first one is the OWL Lite restriction of the second one
201–210	The ontologies under alignment have the same structure, but different lexical and linguistic features
221–247	The ontologies under alignment have the same lexical and linguistic features, but different structure
248–266	The ontologies under alignment have different lexical, linguistic and structure features
301–304	The ontologies under alignment are real world cases

the elite of current generation. First, we calculate each individual's reference individual which is defined as follows: given an individual a , its reference individual $I_{ref}(a)$ is the one that improves a with maximal UIR:

$$I_{ref}(a) = \text{Arg max}(UIR(I, a)) \quad (18)$$

In other words, $I_{ref}(a)$ is better than a with respect to the quality of the alignment of UIR measure. Note that each individual may have one or more reference individuals, and more than one individuals may have the same reference individual. Thus, the more times an individual is a reference individual, the better it will be. Find the individual with the maximum times of being reference individuals, and take it as the best individual. Second, if there are several individuals with the maximal times of being the reference individuals, then a baseline individual, whose weights are all 0.25 and the threshold is 0.80, is taken, and is compared with each of these individuals. The individual with larger UIR is taken as the best individual. Finally, if these individuals still cannot win via UIR, the one with the largest MatchFmeasure will be taken as the best individual.

5.6. Local search process

In general, the local search strategies perform iterative search for optimal solution in the neighborhood of a candidate. In order to tradeoff between the local search and the global search, the local search process in our work is designed according to the following rules:

- the local search is applied within each evolutionary cycle,
- the local search is executed after crossover and mutation,
- the local search is applied to the best individual of population,
- the local search method is the hill climbing algorithm.

In particular, the hill climbing algorithm is a local search iterative method. During iterations, the algorithm attempts to find a better individual by randomly mutating the current one. If a mutation improves the current individual, then the new individual replaces the current one. The search is repeated until no further improvement can be found or after a maximum number of iterations.

Next, we will perform a comparison by experiments among the baseline alignments, which set the weights of each similarity measures as $1/\text{num}$ (num is the number of similarity measures) and the threshold as 0.80, and the alignments obtained by MA using MatchFmeasure only and those by MA using both MatchFmeasure and UIR.

6. Experimental results and analysis

In the experiments, the well-known biblio benchmarks provided by the Ontology Alignment Evaluation Initiative (OAEI) 2013 [14] are used. Each benchmark in the OAEI data set is composed of two ontologies to be aligned and a reference alignment to evaluate the quality of alignment. Moreover, according to OAEI policies, the benchmark reference alignments take into account only matching between ontology classes and properties. In this experiment, we utilize the downloadable datasets from the OAEI 2013 official website for testing purposes. Table 2 shows a brief description about the biblio benchmarks of OAEI 2013.

6.1. Experiments configuration

In the experiments, the similarity measures used are as follows:

- SMOA (Syntactic Measure),
- Linguistic distance based on WordNet (Linguistic Measure),
- Taxonomy distance based on SF algorithm (Taxonomy-based Measure).

The configuration of MA in our work follows the following principles:

- In our work, since the genetic algorithm in MA works mainly based on the crossover operator and is aided by the mutation operator, the crossover possibility should be larger and the mutation possibility just the opposite. However, if the value of the crossover operator is too great, excess solutions would appear which might increase the cost of computation. Therefore, the suggested range of crossover probability is $[0.2, 1]$, and through the preliminary experiment, we find that the results obtained with the crossover probability 0.8 and the mutation probability 0.02 are acceptable for various heterogeneous problem in all testing datasets.
- Since the local searching process requires producing a local searching population with high diversity, the mutation possibility of local search should be higher than that of the genetic algorithm. However, if the value is too large, the produced individual might not be the “neighbor” of the local searching target. Therefore, the suggested range of mutation probability is $[0.2, 0.8]$, and through the preliminary experiment, we find that the mutation probability 0.5 works better.
- The population size, local Search intensity and fitness evaluation number for termination depend on the scale of the problem, the suggested ranges for them are $[30, 300]$, $[10, 50]$ and $[200, 1000]$, respectively. Since the problem scale in our work is not large (only four parameters need to be determined), we set the size of population, maximum number of iterations and the fitness evaluation number for termination as 30, 20 and 250 respectively.

To summary, in our work, MA uses the following parameters which represent a trade-off setting obtained in empirical way to achieve the highest average alignment quality on all test cases of exploited dataset. Through the configuration of parameters chosen in this way, it has been justified by the experiments in this paper that parameters chosen are robust for all the heterogeneous problems presented in the benchmarks, and it is hopeful to be robust for the common heterogeneous situations in the real world.

- Search space for each parameter is the continuous interval $[0, 1]$,
- Population size = 30 individuals,
- Crossover probability = 0.8,
- Mutation probability = 0.02,
- Local Search intensity = 20 iterations,
- Local Search Mutation probability = 0.5,
- Termination condition = 250 fitness evaluations.

The results of the experiments are given in the next section.

6.2. Results and analysis

With respect to aligning one pair of ontologies, Figs. 5–7 show the MatchCoverage (briefly *MC*), Frequency (briefly *F*) and MatchFmeasure (briefly *MF*) values in 20 independent runs, respectively, obtained by the baseline proposal, MA using MatchFmeasure only and MA using both MatchFmeasure and UIR, and Figs. 8 and 9 present UIR values between S_A and S_B and between solutions obtained by baseline proposal and one of S_A and S_B , and Fig. 10 represents the fitness values of the compared methods, where NT represents the number of the test case in horizontal axis and S_A and S_B refer to the solutions obtained by MA using MatchFmeasure only and those by MA using MatchFmeasure and UIR respectively.

With regard to aligning multiple pairs of ontologies, Table 3 shows the average values of *MC*, *F* and *MF* in 20 independent runs (briefly (*MC*, *F*, *MF*)) determined by the baseline proposal, MA using MatchFmeasure only and MA using both MatchFmeasure and UIR. In Table 3, the combinations of benchmarks are selected randomly. Table 4 compares UIR values between two of S_A , S_B and solutions obtained by baseline proposal and gives the average fitness values, where *Base* refers to the solutions obtained by the baseline proposal.

Table 5 shows the ranks of three meta-matching systems, i.e. our proposal, GOAL and ECOMatch, Table 6 presents Holm’s test results and Table 7 gives the comparison of our proposal with the participants in OAEI 2013.

As can be seen from $UIR(S_B, S_A)$ in Fig. 8, S_B is better than S_A in benchmarks 203–207, 209, 230, 239, 240, 246, 247, 252, 254, 261, 266, 301 and 302. In Fig. 9, for those benchmarks whose $UIR(S_B, S_A)$ value is 0, the values of $UIR(S_A, Base)$ and $UIR(S_B, Base)$ distinguish the quality of our solutions on benchmark 208. In Fig. 10, among the rest benchmarks, the values of $fitness(S_A)$ and $fitness(S_B)$ on benchmarks 210, 223, 224, 248, 249, 253, 258 and 265 shows the priority of S_B . During

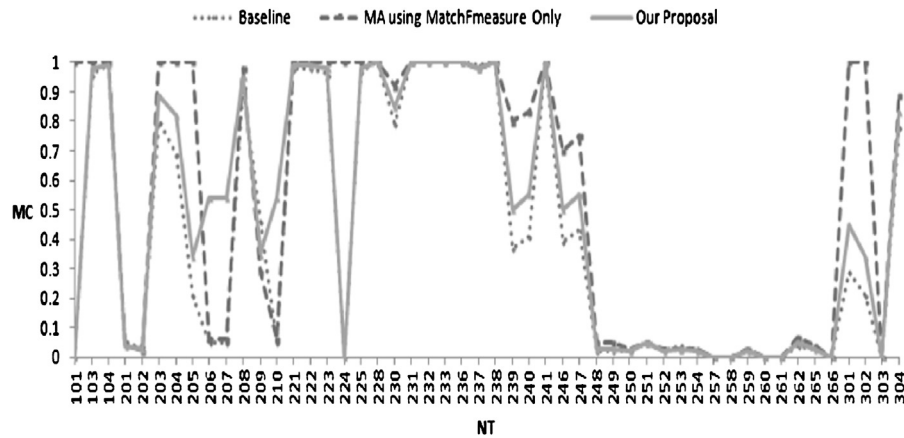


Fig. 5. Comparison of MatchCoverage values obtained by three methods via aligning one pair of ontologies.

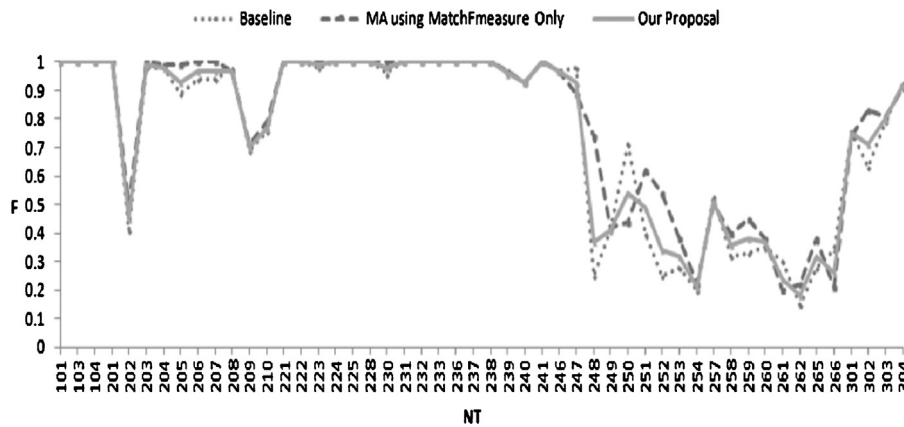


Fig. 6. Comparison of Frequency values obtained by three methods via aligning one pair of ontologies.

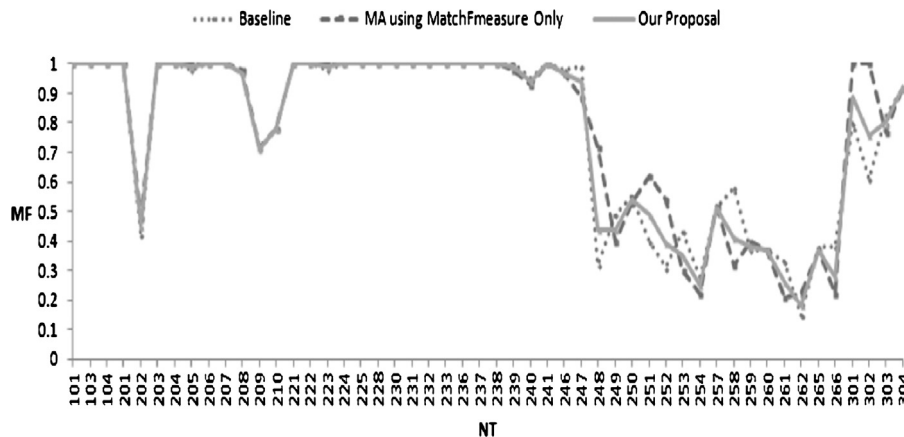


Fig. 7. Comparison of MatchFmeasure values obtained by three methods via aligning one pair of ontologies.

the process of evolution, since our approach utilize UIR to ensure the consistent improvement of recall and precision of the individuals, it is potentially more likely to find better solution than the approach without using UIR. Therefore, through the comparison of the results obtained by one pairs of ontologies using two approaches, MA using both MatchFmeasure and UIR is effective.

As Table 4 shows, except for case number 1, S_B is better than S_A in the rest cases. From the third column, S_B is apparently better than S_A in case numbers 5, 7, 8 and 9. While according to the values in the fourth and fifth columns, S_B is apparently better than S_A in case numbers 2, 3 and 6. Finally, by comparing the fitness of S_B with that of S_A , S_B is

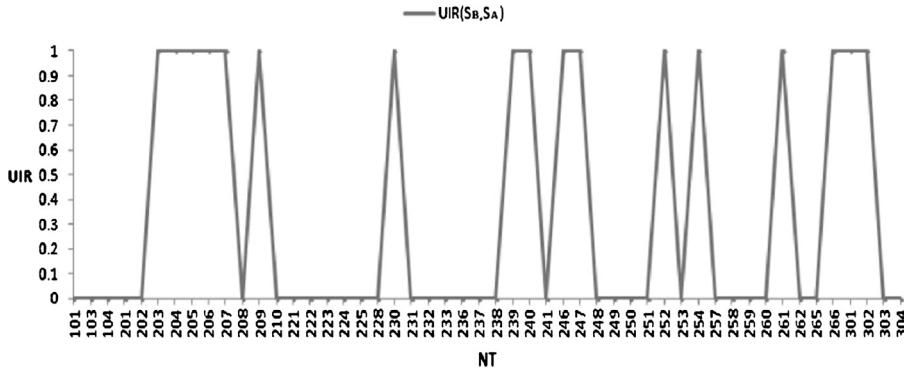


Fig. 8. $U_{IR}(S_B, S_A)$ values via aligning one pair of ontologies.

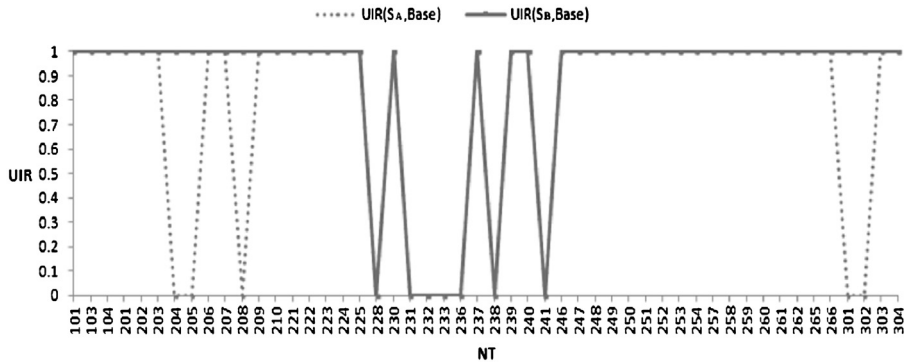


Fig. 9. Comparison of $U_{IR}(S_A, Base)$ and $U_{IR}(S_B, Base)$ values via aligning one pair of ontologies.

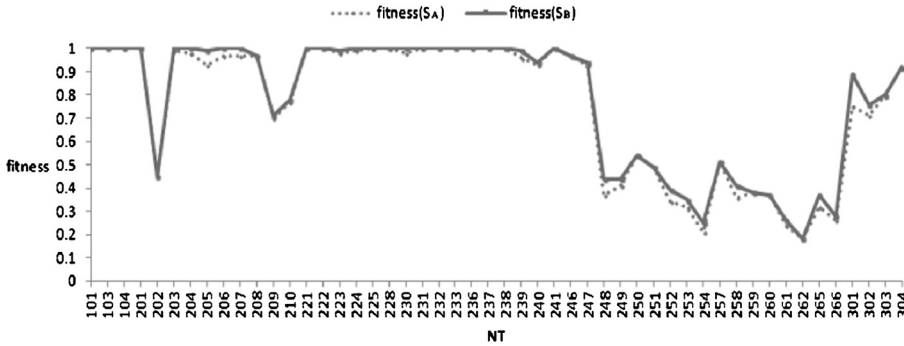


Fig. 10. Comparison of $fitness(S_A)$ and $fitness(S_B)$ values via aligning one pair of ontologies.

better than S_A in case number 4. To sum up, through the comparison of the results obtained by multiple pairs of ontologies using two approaches, MA using both MatchFmeasure and UIR is much better than MA using MatchFmeasure only.

Next, we will describe the statistical comparison carried out between our proposals and the existing meta-matching systems, i.e. GOAL and ECOMatch. In particular, we carried out GOAL with the static threshold 0.8 and ECOMatch using GA with the size of Partial Reference Alignment 75%, and in this experimental scenario, F-measure is used for evaluating the quality of produced alignments. The comparison is formally carried out by means of a multiple comparison procedure which consists of two steps: in the first one, a statistical technique, i.e. Friedman's test, is used to determine whether the results provided by the state of the art ontology systems and our proposal present any difference; in the second one, which method is outperformed is determined by carrying out a post-hoc test, i.e. Holm's test, only if in the first step a difference is found.

Friedman's test is a non-parametric statistical procedure which aims at detecting if a significant difference among the behavior of two or more algorithms exists. In particular, under the null-hypothesis, it states that all algorithms are equivalent, hence, a rejection of this hypothesis implies the existence of differences among the performance of all studied algorithms [17]. In order to reject the null hypothesis, the computed value χ_r^2 must be equal to or greater than the tabled critical chisquare value at the specified level of significance [18]. In our experimentation, a level of significance $\alpha = 0.05$

Table 3

Comparison of (MC, F, MF) values obtained by three methods via aligning multiple pairs of ontologies.

Case	ID	Baseline (MC, F, MF)	MA using MatchFmeasure only (MC, F, MF)	Our approach (MC, F, MF)
1	101	(0.00, 1.00, 0.00)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
	103	(0.96, 1.00, 0.98)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
2	104	(0.99, 1.00, 0.99)	(1.00, 0.98, 0.99)	(1.00, 1.00, 1.00)
	203	(0.80, 1.00, 0.89)	(0.97, 1.00, 0.99)	(0.86, 1.00, 0.92)
3	204	(0.69, 1.00, 0.81)	(0.98, 0.99, 0.98)	(0.94, 0.99, 0.96)
	205	(0.21, 1.00, 0.34)	(0.89, 0.99, 0.93)	(0.45, 1.00, 0.62)
4	204	(0.69, 1.00, 0.81)	(1.00, 1.00, 1.00)	(1.00, 1.00, 1.00)
	221	(0.23, 1.00, 0.37)	(0.49, 0.87, 0.63)	(0.67, 0.73, 0.70)
5	228	(1.00, 1.00, 1.00)	(1.00, 0.89, 0.94)	(1.00, 1.00, 1.00)
	301	(0.29, 1.00, 0.45)	(0.66, 0.75, 0.70)	(0.37, 1.00, 0.54)
6	301	(0.29, 1.00, 0.45)	(0.80, 0.70, 0.75)	(0.41, 0.96, 0.57)
	302	(0.21, 1.00, 0.34)	(0.63, 0.75, 0.68)	(0.40, 1.00, 0.57)
7	101	(0.00, 1.00, 0.00)	(1.00, 0.94, 0.97)	(1.00, 0.98, 0.99)
	221	(0.98, 1.00, 0.99)	(1.00, 0.94, 0.97)	(1.00, 1.00, 1.00)
	301	(0.29, 1.00, 0.45)	(0.64, 0.79, 0.71)	(0.44, 0.96, 0.60)
8	103	(0.96, 1.00, 0.98)	(1.00, 0.98, 0.99)	(1.00, 1.00, 1.00)
	223	(0.96, 1.00, 0.98)	(0.99, 0.94, 0.96)	(0.98, 1.00, 0.99)
	302	(0.21, 1.00, 0.34)	(0.54, 0.90, 0.68)	(0.40, 1.00, 0.57)
9	104	(0.99, 1.00, 0.99)	(1.00, 0.98, 0.99)	(1.00, 1.00, 1.00)
	225	(0.97, 1.00, 0.98)	(1.00, 0.92, 0.96)	(1.00, 1.00, 1.00)
	301	(0.29, 1.00, 0.45)	(0.63, 0.77, 0.69)	(0.24, 1.00, 0.38)

Table 4

Comparison of UIR and fitness values obtained by three methods via aligning multiple pairs of ontologies.

Case	ID	UIR(S_B, S_A)	UIR($S_A, Base$)	UIR($S_B, Base$)	fitness(S_A)	fitness(S_B)
1	101, 103	0.00	1.00	1.00	1.00	1.00
2	104, 203	0.00	0.50	1.00	0.99	0.96
3	204, 205	0.00	0.00	0.50	0.96	0.79
4	204, 221	0.00	0.50	0.50	0.82	0.85
5	228, 301	0.50	−0.50	0.50	0.82	0.77
6	301, 302	0.00	0.00	0.50	0.71	0.57
7	101, 221, 301	0.67	0.00	0.33	0.88	0.86
8	103, 223, 302	0.33	0.00	1.00	0.88	0.85
9	104, 225, 301	0.67	0.00	0.33	0.88	0.79

is chosen. Since in our case we are comparing three ontology matching proposals, our analysis has to consider the critical value $\chi^2_{0.05}$ for two degrees of freedom that is equal to 5.991. In Table 5, in round parentheses, there is the corresponding computed rank for each benchmark.

By performing the Friedman's test, the computed χ^2_r value is 44.53. Since in our case $k = 3$, our analysis has to consider the critical value $\chi^2_{0.05}$ for two degrees of freedom that is equal to 5.991. Since the computed $\chi^2_r = 44.53$ value is greater than its associated critical value $\chi^2_{0.05} = 5.991$, the null hypothesis is rejected and it is possible to assess that there is a significant difference between these proposals.

According to this result, a post-hoc statistical analysis is needed to conduct pairwise comparisons in order to detect concrete differences among compared algorithms. Holm's procedure is a multiple comparison procedure that works by setting a control algorithm and comparing it with the remaining ones. Normally, the algorithm which obtains the lowest value of ranking in the Friedman's test is chosen as control algorithm. In our experimentation, as shown in Table 5, our proposal is characterized by the lowest value of ranking.

Holm's test works on a family of hypotheses where each one is related to a comparison between the control algorithm and one of the remaining algorithms. In details, the test statistic for comparing the i th and j th algorithms named z value is used for finding the corresponding probability from the table of the normal distribution (the so-called p -value), which is then compared with an appropriate level of significance α [19]. In our experimentation $\alpha = 0.05$ and the results of Holm's test are shown in Table 6. By analyzing the data in Table 6, it is possible to state that our proposal statistically outperforms GOAL and ECOMatch at 5% significance level.

Finally, we will compare our proposal with the state of the art ontology alignment systems in terms of F-measure. The compared ontology alignment systems are the participants in OAEI 2013, whose f-measure values of the alignments produced for the benchmark track are from [29]. As can be seen from Table 7, our approach ranks 3rd in all eleven state of the

Table 5

Comparison between existing meta-matching systems and our proposal in terms of F-measure.

Case	Our proposal	GOAL	ECOMatch
101	1.00(2.0)	1.00(2.0)	1.00(2.0)
103	1.00(1.5)	0.99(3)	1.00(1.5)
104	1.00(2.0)	1.00(2.0)	1.00(2.0)
201	1.00(1.0)	0.93(2.0)	0.94(3.0)
202	0.41(1.0)	0.30(3.0)	0.39(2.0)
203	1.00(2.0)	0.98(2.5)	0.98(2.5)
204	1.00(1.5)	1.00(1.5)	0.99(2.0)
205	0.99(1.0)	0.97(2.0)	0.95(3.0)
206	1.00(1.0)	0.96(2.0)	0.92(3.0)
207	1.00(1.0)	0.96(2.5)	0.96(2.5)
208	0.97(1.5)	0.97(1.5)	0.95(3.0)
209	0.67(1.0)	0.64(2.0)	0.59(3.0)
210	0.72(1.0)	0.69(2.0)	0.58(3.0)
221	1.00(1.0)	0.99(2.5)	0.99(2.5)
222	1.00(1.0)	0.98(3.0)	0.99(2.0)
223	0.99(1.0)	0.97(2.0)	0.96(3.0)
224	1.00(2.0)	1.00(2.0)	1.00(2.0)
225	1.00(2.0)	1.00(2.0)	1.00(2.0)
228	1.00(2.0)	1.00(2.0)	1.00(2.0)
230	1.00(1.0)	0.99(2.0)	0.94(3.0)
231	1.00(1.0)	0.99(2.5)	0.99(2.5)
232	1.00(2.0)	1.00(2.0)	1.00(2.0)
233	1.00(1.5)	1.00(1.5)	0.98(3.0)
236	1.00(2.0)	1.00(2.0)	1.00(2.0)
237	1.00(2.0)	1.00(2.0)	1.00(2.0)
238	1.00(1.5)	1.00(1.5)	0.96(3.0)
239	0.99(1.0)	0.96(3.0)	0.98(2.0)
240	0.94(1.0)	0.92(2.0)	0.85(3.0)
241	1.00(1.5)	1.00(1.5)	0.98(3.0)
246	0.97(1.0)	0.93(3.0)	0.94(2.0)
247	0.94(1.0)	0.92(2.0)	0.91(3.0)
248	0.35(1.0)	0.21(3.0)	0.32(2.0)
249	0.40(1.0)	0.36(2.0)	0.32(3.0)
250	0.51(1.0)	0.47(2.0)	0.44(3.0)
251	0.45(1.0)	0.33(3.0)	0.42(2.0)
252	0.34(1.0)	0.22(2.0)	0.32(3.0)
253	0.35(1.0)	0.27(3.0)	0.33(2.0)
254	0.23(1.0)	0.11(3.0)	0.12(2.0)
257	0.51(1.0)	0.49(2.0)	0.32(3.0)
258	0.37(1.0)	0.23(3.0)	0.28(2.0)
259	0.38(1.0)	0.29(2.5)	0.29(2.5)
260	0.37(1.0)	0.31(2.5)	0.31(2.5)
261	0.26(1.0)	0.20(3.0)	0.22(2.0)
262	0.14(1.0)	0.10(2.0)	0.06(3.0)
265	0.35(1.0)	0.17(3.0)	0.27(2.0)
266	0.26(1.0)	0.15(2.0)	0.11(3.0)
301	0.86(1.0)	0.84(2.0)	0.71(3.0)
302	0.73(1.0)	0.58(2.0)	0.41(3.0)
303	0.80(1.0)	0.66(3.0)	0.77(2.0)
304	0.89(1.0)	0.87(2.0)	0.74(3.0)
Avg.	0.76(1.24)	0.72(2.27)	0.71(2.49)

Table 6

Holm's test.

i	System	z value	Unadjust p -value	$\alpha/(k-i)$, $\alpha = 0.05$
2	ECOMatch	4.2500	$4.1045 \times e^{-10}$	0.05
1	GOAL	5.1500	$2.6048 \times e^{-7}$	0.025

art ontology alignment systems. With respect to YAM++ and MAPSSS, although they obtain better results than our approach in terms of f-measure, they require domain specific corpora being the background knowledge for ontology matching, and such documents should be pre-selected before executing the matching systems. Thus, they require more and further information than the proposed approach. More important of all, these documents are not always available, while our approach utilizes common corpora WordNet and suffers no such constraint. In general, we can draw the conclusion that our proposal is effective.

Table 7

Comparison between the state of the art ontology matching systems and our proposal in terms of F-measure.

	edna	Hertuda	HotMatch	LogMap
f-measure	0.48	0.67	0.68	0.64
	LogMapLt	MaasMtch	MapSSS	ServOMap
f-measure	0.66	0.60	0.86	0.67
	WeSeE	Wikimatch	YAM++	Our approach
f-measure	0.69	0.67	0.89	0.76

7. Conclusion

Ontology alignment is an important step in ontology engineering. Although lots of work have been done to tackle this problem, there are still various important issues left for the researchers to deal with. In this paper, a novel approach based on MA using both MatchFmeasure and UIR has been proposed to aggregate different similarity measures into a single metric, and optimize the quality of multiple ontology alignments simultaneously without using reference alignment. The experimental results have shown that the MA using both MatchFmeasure and UIR is effective to automatically configure the parameters of similarity aggregation process and our approach could simultaneously deal with multiple pairs of ontologies and avoid the bias improvement caused by MatchFeasure, and the comparison with state-of-the-art ontology matching systems indicates that our proposal is effective.

In continuation of our research, the work is now being done on embedding MA using both MatchFmeasure and UIR into a real ontology alignment system. We are also interested in developing an Expert Decision Support System to help the ontology alignment system to automatically decide the parameters and even which similarity measures should be utilized. Moreover, since the assumption of complete 1:1 alignments is a rather strong restriction, which might not be suitable for many real-world scenarios, a relaxation of this constraint should be taken into consideration in the future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61272119 and No. 61472297).

References

- [1] J. Euzenat, P. Valtchev, Similarity-based ontology alignment in OWL-Lite, in: *Proceedings of the Sixteenth European Conference on Artificial Intelligence*, 2004, pp. 333–337.
- [2] E. Amigo, J. Gonzalo, J. Artiles, M.F. Verdejo, Combining evaluation metrics via the unanimous improvement ratio and its application to clustering tasks, *J. Artif. Intell. Res.* 42 (2011) 689–718.
- [3] P. Moscato, M.G. Norman, A Memetic approach for the travelling salesman problem – implementation of a computational ecology for combinatorial optimization on message-passing systems, in: *Proceedings of the International Conference on Parallel Computing and Transputer Applications*, 1992, pp. 177–186.
- [4] N.J. Radcliffe, P.D. Surry, Formal memetic algorithms, in: T.C. Fogarty (Ed.), *Evolutionary Computing: AISB Workshop*, in: *LNCS*, vol. 865, Springer-Verlag, 1994, pp. 1–16.
- [5] P. Merz, B. Freisleben, Memetic algorithms for the traveling salesman problem, *Complex Syst.* 14 (2011) 197–345.
- [6] J. Martinez-Gil, E. Alba, J.F. Aldana-Montes, Optimizing ontology alignments by using genetic algorithms, in: *Nature Inspired Reasoning for the Semantic Web, NatuReS2008*, vol. 419, 2008, pp. 31–45.
- [7] J.M.V. Naya, M.M. Romero, J.P. Loureiro, Improving ontology alignment through genetic algorithms, in: *Soft Computing Methods for Practical Environment Solutions: Techniques and Studies*, 2010, pp. 1–16.
- [8] A.-L. Ginsca, A. Iftene, Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment, in: *9th Roedunet Int. Conf., RoEduNet*, 2010, pp. 118–122.
- [9] V. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, *Sov. Phys. Dokl.* 10 (1966) 707–710.
- [10] W. Winkler, The state record linkage and current research problems, Technical report, Statistics of Income Division, Internal Revenue Service Publication, 1999.
- [11] G.A. Miller, WordNet: a lexical database for English, *Commun. ACM* 38 (1995) 39–41.
- [12] C.J. Van Rijsbergen, Foundation of evaluation, *J. Doc.* 34 (1974) 365–373.
- [13] F.T. Fang, Y. Wang, *Number-Theoretic Methods in Statistics*, Chapman & Hall, London, UK, 1994.
- [14] Ontology alignment evaluation initiative (OAEI), available at <http://oei.ontologymatching.org/2013/>, Accessed July 15, 2014.
- [15] G. Stoilos, G. Stamou, S. Kollias, A string metric for ontology alignment, in: *Proceedings of 4th International Semantic Web Conference, ISWC 2005*, 2005, pp. 623–637.
- [16] T. Kirsten, A. Thor, E. Rahm, Instance-based matching of large life science ontologies, in: *DILS'07 Proceedings of the 4th International Conference on Data Integration in the Life Sciences*, 2007, pp. 172–187.
- [17] S. Garcia, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, *J. Heuristics* 15 (2009) 617–644.
- [18] D.J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall/CRC, 2000.
- [19] J. Demsar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (2006) 1–30.
- [20] J. Bock, J. Hettenhausen, Discrete particle swarm optimisation for ontology alignment, *Inf. Sci.* 192 (2012) 152–173.
- [21] G. Acampora, U. Kaymak, V. Loia, et al., Applying NSGA-II for solving the ontology alignment problem, in: *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2013, pp. 1098–1103.

- [22] E. Rahm, P.A. Bernstein, A survey of approaches to automatic schema matching, *VLDB J.* 10 (4) (2001) 334–350.
- [23] S. Melnik, H. Garcia-Molina, E. Rahm, Similarity flooding: a versatile graph matching algorithm and its application to schema matching, in: 18th International Conference on Data Engineering, 2002, pp. 117–128.
- [24] D. Ritze, H. Paulheim, Towards an automatic parameterization of ontology matching tools based on example mappings, in: Proceedings of the Sixth International Workshop on Ontology Matching at ISWC, vol. 814, 2011, p. 37.
- [25] G. Stumme, M. Ehrig, S. Handschuh, et al., The Karlsruhe view on ontologies, Technical report, University of Karlsruhe, Institute AIFB, 2003.
- [26] J. Euzenat, P. Shvaiko, *Ontology Matching*, Springer, Heidelberg, 2007.
- [27] E. Amigo, J. Artilles, J. Gonzalo, Combining evaluation metrics with a unanimous improvement ratio and its application to the web people search clustering task, in: Proceedings of the 2nd Web People Search Evaluation Workshop, WePS 2009, 2009.
- [28] Y.S. Ong, N. Krasnogor, H. Ishibuchi, Special issue on memetic algorithms, *IEEE Trans. Syst. Man Cybern., Part B, Cybern.* 37 (1) (2007) 2–5.
- [29] B.C. Grau, Z. Dragisic, K. Eckert, et al., Results of the ontology alignment evaluation initiative 2013, in: Proceedings of 8th ISWC Workshop on Ontology Matching, 2013.
- [30] A. Vitiello, G. Persiano, V. Loia, G. Acampora, Memetic algorithms for ontology alignment, Università degli Studi di Salerno, Italy, 2013.