Optimizing Ontology Alignments by using NSGA-II

Xingsi Xue, Yuping Wang and Weichen Hao
School of Computer Science and Technology, Xidian University, China

Abstract: *In this paper, we propose a novel approach based on NSGA-II to address the problem of optimizing the aggregation of three different basic similarity measures (syntactic measure, linguistic measure and taxonomy-based measure) and get a single similarity metric. Comparing with conventional genetic algorithm, the proposed method is able to realize three goals simultaneously, i.e., maximizing the alignment recall, the alignment precision and the F-measure and find the optimal solutions which could avoid bias to recall or precision value. Experiment results show that the proposed approach is effective.*

Keywords: *Ontology alignment, NSGA-II, aggregation of similarity measures.*

Received June 15, 2012; accepted September 26, 2014; published online April 23, 2014

1. Introduction

Since, the ontology allows data and knowledge to be shared and reused more effectively, it is widely used in information exchange between heterogeneous data sources in semantic web. However, because of human subjectivity, various ontologies related to the same application domain may define one entity with different names or in different ways, raising so-called heterogeneity problem. Addressing this problem requires to identify correspondences between the entities of various ontologies. This process is commonly known as ontology alignment.

It is highly impractical to align the ontologies manually when the size of ontologies is considerable large. Thus, numerous alignment 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 elements are semantically similar or not. Since, none of the similar measures could provide the satisfactory result independently, most ontology alignment systems combine a set of different similar measures together by aggregating their aligning results. How to select the appropriate similar 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 techniques like genetic algorithms. However, current meta-matching approaches generally determine the weights by the single objective approach which may lead to unwanted bias to one of the evaluations of the alignment quality. Given a set of existing ontology alignments (reference alignments), the aim of this paper utilize NSGA-II to find the global non-dominated set of parameters, such as weights and thresholds, to combine multiple similarity measures into a single aggregated metric, In this way, we could realize three goals simultaneously, i.e., maximizing the alignment precision, the alignment recall and the F-measure and find the optimal solutions which could avoid bias to recall or precision value.

NSGA-II is considered to be a flexible and robust technique, which is good at finding various non-dominated solutions quickly. First, the algorithm applies the standard crossover and mutation operators in the evolution of current population. Then, it uses the fast non-dominated sorting technique and a crowding distance to rank and select the next generation. Finally, the best individuals in terms of non-dominance and diversity are selected as the solutions. Therefore, it's apparently suitable to utilize NSGA-II to aggregate different similarity values and get various global non-dominated optimal alignments.

The rest of the paper is organized as follows: Section 2 is devoted to present state of the art about some important ontology alignment systems, section 3 provides a detailed description of the basic concepts of ontology and ontology alignment, section 4 proposes the framework of using NSGA-II to solve the ontology alignment problem, in section 5, the experimental results show the results of our approach for solving the alignment problem; finally, in section 6, we draw conclusions and propose further improvement.

2. Related Work

In recent years, numerous fully automatic or semi-automatic matching systems have been developed. The first ones used only one or few alignment approaches. However, because of the heterogeneity and ambiguity of data description, it is unavoidable that optimal mappings for various pairs of entities will be considered as "best mappings" by none of the existing ontology alignment approaches. For this reason, it is necessary to compose these simple approaches. The next generation of matching systems combines more diverse similar measures to determine correspondences between ontology elements. The most outstanding approaches in this area are COMA [5], COMA++ [2], QuickMig [6] and OntoBuilder [10], but these systems use weights determined by an expert. Lately, the focus is on meta-matching. Meta-matching does not use parameters from an expert, but selects those 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 genetic algorithms.

Among meta-matching systems that make use of a genetic algorithm, the most notable one is Genetics for Ontology Alignments (GOAL) [15]. GOAL does not directly compute the alignment between two ontologies, but it determines, through a genetic algorithm, 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 two more recent papers [12, 19]. All of the systems mentioned above work with only one of several common measures that used to evaluate the quality of an alignment. However, these measures could simply evaluate the aligning results in one aspect, respectively. Therefore, the current approaches cannot satisfy multifarious requirements of alignments. Our work is to utilize the NSGA-II algorithm in the whole similarity aggregation step of meta-matching system, to provide the diverse global non-dominated sets of weights and thresholds for meta-matching system to meet the diverse requirements of alignment.

3. Related Concepts

3.1. Ontology and Ontology Alignment

There are many 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 convenience of the work in this paper, an ontology can be defined in definition 1 [1].

Definition 1. An ontology is a triple O=(C, P, I), where:

C: Is the set of classes, i.e., the set of concepts that populate the domain of interest.
P: Is the set of properties, i.e., the set of relations existing between the concepts of domain.
I: Is the set of individuals, i.e., the set of objects of the real world, representing the instances of a concept.

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

Ontologies are seen as the solution to data heterogeneity on the web. However, the existing ontologies could themselves introduce heterogeneity: Given two ontologies, the same entity can be given different names or simply be defined in different ways, whereas both ontologies may express the same knowledge but in different languages [9]. To solve this problem, a so-called ontology alignment process is necessary. Formally, an alignment between two ontologies can be defined as presented by Definition 2 [1].

Definition 2. An alignment between two ontologies is a set of mapping elements. A mapping element is a 4-uple $\left(e,e',n,r\right)$, where:

e and e': Are the entities of the first and the second ontology, respectively.
n: Is a confidence measure in some mathematical structure (typically in the (0, 1) range) holding for the correspondence between the entities e and e'.
R: Is a relation (typically the equivalence) holding between the entities e and e'.

The ontology alignment process can be defined as follows [1]:

Definition 3. The alignment process can be seen as a function □ which, from a pair of ontologies O and O' to align, an input alignment A, a set of parameters p, a set of resources r, returns a new alignment A' between these ontologies: A'=□(O, O', A, p, r).

The ontology alignment process computes a mapping element by using a similarity measure, which determines the closeness value n (related to a given relation R) between the entities e and e' in the range (0, 1), where 0 stands for complete inequality and 1 for complete equality.

Next, we describe a general classification of the most used similarity measures.

3.2. Similarity Measures

Typically, similarity measures could be categorized in syntactic, linguistic and taxonomy-based measures. In the following, we present some common similarity measures belonging to these three categories.

3.2.1. Syntactic Measures

Syntactic measures compute a string distance or edit distance between the ontology entities. In our work, we utilize two widely used syntactic measures: Levenstein distance [14] and Jaro distance [8].

Levenstein distance calculates the number of operation, such as modification, deletion and insertion of a character. To do so, it is necessary to transform one string into another. Formally, the Levenstein distance between two strings s_1 and s_2 is defined by the following equation:

$$Levenstein(s_1, s_2) = max\ (0, \frac{min(|s_1|, |s_2|) - d(s_1, s_2)}{min(|s_1|, |s_2|)}) \qquad (1)$$

Where:

$|s_1|$ and $|s_2|$: Is the length of string s_1 and s_2, respectively.
$d(s_1, s_2)$: Is the number of operation necessary to transform s_2 into s_2.

Another measure is the Jaro distance, an edit distance that uses the number of common characters in the two strings and the positions in which they appear. Given strings s_1 and s_2, the Jaro distance is defined as follows:

$$JaroDist(s_1, s_2) = \frac{1}{3}\left(\frac{com(s_1,s_2)}{|s_1|} + \frac{com(s_1,s_2)}{|s_2|} + \frac{com(s_1,s_2) - trans(s_1,s_2)}{com(s_1,s_2)}\right) \quad (2)$$

Where:

$|s_1|$ and $|s_2|$: Is the length of string s_1 and s_2, respectively.
$com(s_1, s_2)$: Is the number of common characters of s_1 and s_2.
$trans(s_1, s_2)$: Is the number of pairs consisting of common characters that appear in different positions.

3.2.2. Linguistic Measures

Linguistic measure calculates the similarity between ontology entities by considering linguistic relations such as synonymy, hypernym and so on. In the proposed work, WordNet [18], which is an electronic lexical database where various senses of words are put together into sets of synonyms, is used to calculate a synonymy-based distance by considering the name of entities. Given two words w_1 and w_2, $LinguisticDist(w_1, w_2)$ equals:

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

3.2.3. Taxonomy-based Measures

Taxonomy-based measures consider only the specialization relation. The intuition behind taxonomic measures is that subsumption relation connect terms that are already similar, therefore, their neighbors may be also somehow similar. For instance, if super-concepts are the same, the actual concepts are similar to each other; if sub-concepts are the same, the compared concepts are also similar. Formally, let c_1 and c_2 be classes of two ontologies O_1 and O_2, s_1 and s_2 be superclasses or subclasses of c_1 and c_2, respectively. There is a correspondence $c=(s_1, s_2)$ with an evaluation $f(c)$, then $TaxonomyDistance(c_1, c_2)=f(c)$.

To combine all the similarity measures mentioned above, an aggregation strategy is needed. In this work, we utilize weighted average aggregation which is defined in the following:

$$\phi(\vec{s}(c),\vec{w}) = \sum_{i=1}^{n} w_i s_i(c) \, with \sum_{i=1}^{n} w_i = 1 \, and \, w_i \in [0,1] \quad (3)$$

Where:

$\vec{s}(c)$: Is the vector of similarity measure results.
$\vec{w}$: Is the vector of weights.
n: 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 derive from the information retrieval field [22] in the next section.

3.3. Alignment Evaluation

The alignment is normally assessed on the basis of two measures commonly known as recall and precision. Recall (or completeness) measures the fraction of correct alignments found in comparison to the total number of correct existing alignments. A recall of 1 means that all of the alignments have actually been found, but it does not provide the information about the number of additionally falsely identified alignment. Typically, recall is balanced against precision (or correctness), which measures the fraction of found alignments that are actually correct. A precision of 1 means that all found alignments are correct, but it does not imply that all alignments have been found. Therefore, recall and precision are often balanced against each other with the so-called F-measure, which is the uniformly weighted harmonic mean of recall and precision. However, when two alignments' F-measure is equal, it's difficult to say which one is better or has less bias to recall or precision.

Given a reference alignment R and some alignment A, recall, precision and F-measure are given by the following formulas:

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

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

$$f - measure = 2 \times \frac{precision \times recall}{precision + recall} \quad (6)$$

In our work, recall and precision of the alignment are taken as two objectives of the meta-matching problem and we intend to maximize both of them. In order to, find diverse global non-dominated optimal solutions of the problem, we utilize NSGA-II which will be discussed in details in section 4.

4. NSGA-II For Ontology Alignments

There are some preparation steps before deploying the NSGA-II. First, the similarity measures are chosen. Second, given two ontologies as the input, the values of these measures are calculated and the results are stored in XML format. This is done to avoid recalculating the similarity during the process of running NSGA-II.

Finally, we calculate an aggregated similarity using the aggregation strategy defined in 3.2.3. In the following, four basic steps of NSGA-II are presented.

4.1. Chromosome Encoding

We incorporate in a chromosome both the weights associated with the similarity measures and the threshold to decide whether a pair of entities is an alignment or not. Therefore, one chromosome can be divided into two parts, one stands for several weights and the other for threshold. Concerning the characteristics of the weights which are mentioned in 3.2.3, our encoding mechanism indirectly represents them by defining the cut or separation point in the interval [0, 1] that limits the value of the weights. If p is the number of weights required, the set of cuts can be represented as $c' = \{c_1', c_2', \ldots, c_{p-1}'\}$. The chromosome decoding is carried out by queuing the elements of c' in ascending order, then we get $c = \{c_1, c_2, \ldots, c_{p-1}\}$ and calculating the weights as follows:

$$w_k = \begin{cases} c_1 & k=1 \\ c_k - c_{k-1} & 1< k< p \\ 1 - c_{p-1} & k=p \end{cases} \qquad (7)$$

Therefore, the length of a chromosome is *(n-1)×cutLength+thresholdLength*, where n is the number of weights, *cutLength* and *thresholdLength* are the chromosome lengths of the cut and threshold, respectively.

4.2. Fitness Functions

Fitness functions are objective functions that evaluate the quality of the alignment obtained by using the weights and the threshold encoded in the chromosome. In our work, there are two objective functions calculating the recall and precision value of the aggregating result, respectively.

4.3. Genetic Operators

4.3.1. Selection

In order to ensure the diversity of the population and accelerate the convergence of the algorithm, selection operator first queues the chromosomes of population in descending order according to their crowding distances which estimate the density of the solutions. Then, we select half of the chromosomes in the front of the population and randomly copy one each time until forming a new population.

4.3.2. Crossover

The crossover operator takes two chromosomes called parents and generates two children chromosomes, which are obtained by mixing the genes of the parents. Crossover is applied with a certain probability, a parameter of the genetic algorithm. In this work, we use the common one-cut-point method to carry out the crossover operation on the population. 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.

4.3.3. Mutation

Mutation operator assures diversity in the population and prevents premature convergence. In our work, for each bit in the chromosome we check if the mutation could be applied according to the mutation probability and if it is, the value of that bit is then flipped.

4.4. Generation of the Next Generation Population

First, we put the current population and the new population together and remove the redundancy of the chromosomes. Then, the new population is selected by non-dominated-sorting and the crowd-distance which is presented in details in [4].

When the algorithm terminates, we propose a selecting strategy to select the representative solutions, i.e., select those with the best recall or precision or F-measure, from the first front. Concretely, for the solutions with the best recall, we will select one solution which has the highest precision. Similarly, for the solutions with the highest precision, we will select one solution with the best recall. Among the solutions with the highest F-measure, we adopt the max-min approach to get a better solution, i.e., suppose that solutions $x_1, x_2, \ldots, x_k$ have the highest F-measure and their recall and precision values are denoted by $f_r(x_k)$ and $f_p(x_k)$, respectively, for $i = 1 \sim k$. Then, we select the solution by the max-min approach as follows:

$$x_j = \arg\max_i \left\{ \min \left\{ f_r(x_i), f_p(x_i) \right\} \right\} \qquad (8)$$

In the following, we take an example to illustrate the procedure of max-min approach. For instance, there are two solutions with the same F-measure of 0.97 while the recall and the precision of the first solution is 0.95 and 1.0 respectively, and the recall and the precision of the second solution is 0.98 and 0.97 respectively. First, we select the smaller value of recall and precision in the first solution, which is 0.95 and then the smaller one in the second solution, which is 0.97. Since, 0.97 is larger than 0.95, the second solution is better than the first one, which means the solution has less bias to recall and precision than the first one.

Next, we will perform a comparison by experiments between the alignments obtained by using the conventional genetic algorithm with elitism strategy and by utilizing our approach.

5. Experimental Results and Analysis

In the experiments, the well-known benchmarks provided by the Ontology Alignment Evaluation Initiative (OAEI) [20] 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 the matching between ontology classes and properties. Table 1 shows a brief description about the benchmarks of OAEI 2011.

Table 1. Brief description of benchmarks.

ID	Brief Description
101	Strictly Identical Ontologies
103	A Regular Ontology and Other with A Language Generalization
104	A Regular Ontology and Other with A Language Restriction
201	Ontologies without Entity Names
203	Ontologies without Entity Names and Comments
204	Ontologies with Different Naming Conventions
205	Ontologies Whose Labels are Synonymous
206	Ontologies Whose Labels are in Different Languages
221	A Regular Ontology and Other with No Specialisation
222	A Regular Ontology and Other with A Flattened Hierarchy
223	A Regular Ontology and Other with A Expanded Hierarchy
224	Identical Ontologies without Instances
225	Identical Ontologies without Restrictions
228	Identical Ontologies without Properties
230	Identical Ontologies with Flattening Entities
231	Identical Ontologies with Multiplying Entities
301	A Real Ontology About Bibliography Made by MIT
302	A Real Ontology with Different Extensions and Naming Conventions

5.1. Experiments Configuration

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

- Levenstein Distance (Syntactic Measure).
- Jaro Distance (Syntactic Measure).
- Linguistic Distance (Linguistic Measure).
- Taxonomy Distance (Taxonomy-Based Measure).

The conventional genetic algorithm and NSGA-II use the following parameters:

- Search space for each parameter is the continuous interval(0, 1).
- Numerical accuracy=0.01.
- The fitness of conventional genetic algorithm can be recall or precision or F-measure, while the fitnesses of NSGA-II are recall and precision.
- Population size=20 chromosomes.
- Crossover probability=0.6.
- Mutation probability=0.01.
- Max generation=5. After ten independent executions, we noticed that the genetic algorithm does not improve the results beyond the fifth generation, so we have set a limit of five generations.

The hardware configurations used to run the algorithms are provided below:

- Processor: Intel Core (TM) i7.
- CPU speed: 2.93GHz.
- RAM capacity: 4GB.

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

5.2. Results and Analysis

Tables 2 and 3 show the average value obtained by the Recall driven, Precision driven, F-measure driven genetic algorithm and NSGA-II in ten independent runs respectively. Table 2 gives the results of Recall driven, Precision driven genetic algorithm and NSGA-II, where the second and fourth columns show the results of the recall driven and precision driven genetic algorithm respectively and the third and fifth columns give the best recall and best precision of NSGA-II respectively. Table 3 gives the F-measure obtained by F-measure driven genetic algorithm and NSGA-II. In Tables 2 and 3, symbols *R* and *P* stand for recall and precision values respectively.

Table 2. Comparison of the recall and the precision obtained by genetic algorithm and NSGA-II

ID	R(P) (GA)	R(P) (NSGA-II)	P(R) (GA)	P(R) (NSGA-II)
101	1.00 (0.78)	1.00 (1.00)	1.00 (0.01)	1.00 (1.00)
103	1.00 (0.68)	1.00 (1.00)	1.00 (0.98)	1.00 (1.00)
104	1.00 (0.65)	1.00 (1.00)	1.00 (0.99)	1.00 (1.00)
201	0.95 (0.04)	0.98 (0.03)	1.00 (0.01)	1.00 (0.31)
203	1.00 (0.61)	1.00 (0.80)	1.00 (0.83)	1.00 (0.98)
204	1.00 (0.13)	1.00 (0.23)	1.00 (0.74)	1.00 (0.93)
205	0.98 (0.03)	0.98 (0.03)	1.00 (0.21)	1.00 (0.48)
206	0.72 (0.03)	0.73 (0.03)	1.00 (0.23)	1.00 (0.23)
221	1.00 (0.52)	1.00 (1.00)	1.00 (0.99)	1.00 (1.00)
222	1.00 (0.75)	1.00 (1.00)	1.00 (0.99)	1.00 (1.00)
223	1.00 (0.27)	1.00 (0.78)	1.00 (0.96)	1.00 (0.98)
224	1.00 (0.63)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
225	1.00 (0.75)	1.00 (1.00)	1.00 (0.97)	1.00 (1.00)
228	1.00 (0.68)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
230	1.00 (0.54)	1.00 (1.00)	1.00 (0.97)	1.00 (1.00)
231	1.00 (0.65)	1.00 (1.00)	1.00 (0.98)	1.00 (1.00)
301	0.95 (0.03)	1.00 (0.02)	1.00 (0.30)	1.00 (0.39)
302	0.91 (0.02)	1.00 (0.02)	1.00 (0.25)	1.00 (0.40)

Table 3. Comparison of the F-measure obtained by genetic algorithm and NSGA-II

ID	F-measure(R, P) (GA)	F-measure(R, P) (NSGA-II)
101	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
103	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
104	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
201	0.94 (0.90, 0.98)	0.94 (0.90, 0.98)
203	0.99 (0.98, 1.00)	0.99 (0.98, 1.00)
204	0.98 (0.99, 0.98)	0.98 (0.99, 0.98)
205	0.89 (0.90, 0.89)	0.94 (0.89, 0.99)
206	0.70 (0.67, 0.73)	0.70 (0.67, 0.73)
221	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
222	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
223	0.99 (0.98, 1.00)	0.99 (0.98, 1.00)
224	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
225	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
228	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
230	0.99 (1.00, 1.00)	1.00 (1.00, 1.00)
231	1.00 (1.00, 1.00)	1.00 (1.00, 1.00)
301	0.75 (0.73, 0.77)	0.75 (0.75, 0.75)
302	0.71 (0.61, 0.84)	0.71 (0.62, 0.83)

It can be seen from Table 2, the best recall results of NSGA-II are better than those of recall driven genetic algorithm in all benchmarks except 205 whose alignment's quality is equal. For instance, with regard to benchmark 201, the recall value obtained by NSGA-II is higher than that given by the Recall driven genetic algorithm; while in benchmark 222, although the recall values are equal, the precision value obtained by NSGA-II is higher than that given by the Recall driven genetic algorithm. Nevertheless, the best precision results of NSGA-II are better than those of the Precision driven genetic algorithm in all benchmarks except 206, 224 and 228 whose alignment's qualities are equal. For example, with regard to benchmark 103, although the precision values are the same, the recall value obtained by NSGA-II is higher than that given by the Precision driven genetic algorithm. Since, NSGA-II take both the recall and precision into consideration, it's more likely to provide the better solution than the Recall driven or the Precision driven genetic algorithm which consider recall or precision only.

In Table 3, as it can be seen that, the results obtained by the F-measure driven genetic algorithm and NSGA-II are the same except the benchmark 205, 301 and 302. In benchmark 205, the F-measure value provided by NSGA-II is higher than that given by the F-measure driven genetic algorithm. While judging by the max-min approach presented in section 4.4, the results obtained by NSGA-II is better than those given by the F-measure driven genetic algorithm in benchmark 301 and 302.

To conclude, in the process of optimizing ontology alignments, NSGA-II is able to find optimal solutions which are equal to or better than the results obtained by conventional genetic algorithm with elitism strategy. Due to the approach of generating the new generation population in NSGA-II, which ensures the consistent improvement both for recall and precision, those solutions in the first non-dominated front are apparently better than the others in terms of both recall and precision. Therefore, NSGA-II increases the chances of finding better solutions than conventional genetic algorithm in the problem of optimizing ontology alignments.

6. Conclusions

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. One of these issues is the aggregation of different similarity measures into a single similarity metric. We formulate the aggregating process as an optimization problem which can be solved by heuristic techniques such as genetic algorithm.

In the proposed work, a novel approach based on NSGA-II has been proposed to aggregate different similarity measures into a single metric and optimize the quality of the alignment results. The experiment results have shown that the proposed approach using NSGA-II is effective to automatically configure the parameters of similarity aggregation process and our approach could find the optimal solutions which equal to or better than the results from conventional genetic algorithm with elitism strategy.

In continuation of our research, work is now being done on embedding NSGA-II into a real ontology alignment system. We are also interested in developing an expert decision support system to help the ontology alignment system automatically decide the parameters and even which similarity measures should be utilized.

Acknowledgements

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

References

[1] Acampora G., Loia V., and Salerno S., "A Hybrid Evolutionary Approach for Solving the Ontology Alignment Problem," *International Journal of Intelligent Systems*, vol. 27, no. 3, pp. 189-216, 2012.

[2] Aumueller D., Do H., and Massmann S., "Schema and Ontology Matching with COMA++," *in Proceedings of the International Conference on Management of Data and Symposium on Principles Database and Systems*, USA, pp. 906-908, 2005.

[3] Chen J., Chen M., and Huang Y., "Concept Feature-based Ontology Construction and Maintenance," *in Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology*, Chengdu, China, pp. 28-32, 2010.

[4] Deb K., Agrawal S., and Pratap A., "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," *in Proceedings of the Parallel Problem Solving from Nature Conference*, Paris, France, pp. 849-858, 2000.

[5] Do H. and Rahm E., "COMA-a System for Flexible Combination of Schema Matching Approaches," *in Proceedings of the 28th International Conference on Very Large Databases*, Hong Kong, China, pp. 610-621, 2002.

[6] Drumm C., Schmitt M., and Do H., "Quickming: Automatic Schema Matching for Data Migration Projects," *in Proceedings of the 6th ACM Conference on Conference on Information and Knowledge Management*, Salzburg, Austria, pp. 107-116, 2007.

[7] Eckert K., Meilicke C., and Stuckenschmidt H., "Improving Ontology Matching using Meta-level Learning,“ *in Proceedings of Semantic Web Conference: Research and Applications*, Greece, pp. 158-172, 2009.

[8] Euzenat J. and Shvaiko P., *Ontology Matching*,

Springer, Heidelberg, German, 2007.

[9] Euzenat J. and Valtchev P., "Similarity-Based Ontology Alignment in OWL-Lite," *in Proceedings of the 16th European Conference on Artificial Intelligence Proceedings*, Valencia, Spain, pp. 333-337, 2004.

[10] Gal A., Anaby-Tavor A., and Trombetta A., "A Framework for Modeling and Evaluating Automatic Semantic Reconciliation," *Very Large Databases Journal*, vol. 14, no. 1, pp. 50-67, 2005.

[11] Ghobadi A. and Rahgozar M., "An Ontologybased Semantic Extraction Approach for B2C eCommerce[J]," *the International Arab Journal of Information Technology*, vol. 8, no. 2, pp. 163-170, 2011.

[12] Ginsca L. and Iftene A., "Using a Genetic Algorithm for Optimizing the Similarity Aggregation Step in the Process of Ontology Alignment," *in Proceedings of the 9th Roedunet International Conference*, Sibiu, Romania, pp. 118-122, 2010.

[13] Huang Q., Buckley B., and Kechadi M., "Multi-Objective Feature Selection by using NSGA-II for Customer Churn Prediction in Telecommunications," *Expert Systems with Applications*, vol. 37, no. 5, pp. 3638-3646, 2010.

[14] Maedche A. and Staab S., "Measuring Similarity between Ontologies," *in Proceedings of the International Conference on Knowledge Engineering and Knowledge Management*, Sig enza, Spain, pp. 251-263, 2002.

[15] Martinez-Gil J., Alba E., and Aldana-Montes F., "Optimizing Ontology Alignments by using Genetic Algorithms," *Nature Inspired Reasoning for the Semantic Web*, vol. 419, pp. 31-45, 2008.

[16] Martinez-Gil J. and Aldana-Montes F., "Evaluation of Two Heuristic Approaches to Solve the Ontology Meta-Matching Problem," *Knowledge and Information Systems*, vol. 26, no. 2, pp. 225-247, 2011.

[17] Mazak A., Schandl B., and Lanzenberger M., "Align++ A Heuristic-based Method for Approximating the Mismatch-at-risk in Schema-based Ontology Alignment," *in Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, Valencia, Spain, pp. 17-26, 2010.

[18] Miller A., "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39-41, 1995.

[19] Naya V., Romero M., and Loureiro P., *Soft Computing Methods for Practical Environment Solutions: Techniques and Studies*, IGI Group, New York, USA, 2010.

[20] Ontology Alignment Evaluation Initiative (OAEI)., available at: http://oaei.ontologymatching.org/2011/, last visited 2012.

[21] Shvaiko P. and Euzenat J., "A Survey of Schema-based Matching Approaches," *the Journal on Data Semantics, Lecture Notes in Computer Science*, vol. 3730, pp. 146-171, 2005.

[22] Van Rijsbergen J., *Information Retrieval*, Butterworth, London, 1975.

Xingsi Xue is a Lecturer at Fujian University of Technology and a PhD student in computer applications at School of Computer Science and Technology, Xidian University, China. His research interests include ontology matching technology, intelligent expert decision system and object-oriented technology.

Yuping Wang is a professor and PhD supervisor at School of Computer Science and Technology, Xidian University, China. He received his PhD degree from the Department of Mathematics，Xi'an Jiaotong University, China in 1993, Currently, his research interests include optimization methods, theory and application, evolutionary computation, data mining and machine learning.

Weichen Hao is a Master student at School of Computer Science and Technology. His research interests include software engineering, ontology matching technology and data mining.