Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **3**, May **2017**

An Evolutionary Algorithm based Ontology Matching System

Xingsi Xue* and Zhengyi Tang

College of Information Science and Engineering
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
No3 Xueyuan Road, University Town, Minhou, Fuzhou City, Fujian Province, 350118, China
*Corresponding author:jack8375@gmail.com

Received July, 2016; revised December, 2016

Abstract. *Due to the high heterogeneity of ontologies, a combination of many methods is necessary to discover correctly the semantic correspondences between their elements. An ontology matching system can be seen as a collection of several matching components, which implements a specific method dealing with a specific heterogeneity type like the terminological, structural and semantic matchers. In this paper, a novel ontology matching system architecture is proposed and the details of its kernel model are presented, i.e. matching and mapping extraction module, where we construct a new optimal model for the ontology matching and mapping extraction problem and design a problem specific Evolutionary Algorithm (EA) for it. The experimental results show that the mean recall and precision of our approach is generally high, and the mean f-measure of the alignments obtained by our approach outperform all other state of the art matching systems.*
Keywords: ontology matching system, Evolutionary Algorithm, matcher combination, mapping extraction

1. **Introduction.** Once the semantic heterogeneity problems severely hamper different ontologies from communicating, the aim of implementing the cooperation between the application built on these ontologies might not be achieved. To deal with the semantic heterogeneity problem in ontology engineering, recently, a lot of ontology matching systems have been developed by taking into account various aspects of this problem. Methodologically speaking, these approaches rely on techniques from fields as diverse as machine learning, graph matching, information retrieval, relational algebra, logics and so forth, and each of them provides a framework to deal with a certain heterogeneity type. Due to the high heterogeneity of ontologies, a combination of many methods is necessary to correctly determine the semantic correspondences between ontology elements [1]. In this work, we contribute to determine the optimal solution of matcher combination and mapping extraction problem, which is the fundamental for the development of a stable ontology matching system. Currently, the most outstanding approaches in this area are COMA [2], COMA++ [3], QuickMig [4] and OntoBuilder [5], but these systems use the aggregating weights determined by an expert. Later, there emerges a heuristic meta-matching system, which 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. Since modeling the matching and mapping extraction problem is a complex (nonlinear problem with many local optimal solutions) and time-consuming task

(large scale problem), approximate methods are usually used for computing the parameters. From this point of view, Evolutionary Algorithm (EA) [6] could represent an efficient approach for addressing this problem. Among the heuristic meta-matching systems based on EA, the most notable one is GOAL (genetics for ontology alignments) [7]. 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. Our system is also a heuristic meta-matching system, however, different from existing EA based heuristic meta-matching system, we propose a novel system architecture and construct a new optimal model for the matcher combination and mapping extraction problem and design a problem-specific EA to solve it.

The rest of the paper is organized as follows: Section 2 describes the proposed ontology matching architecture. Section 3 presents the matching and mapping extraction module of our ontology matching system, Section 4 gives the experimental results, and Section 5 concludes the paper.

2. **Ontology Matching Architecture.** The main components of a stand alone ontology matching system are depicted in the Figure 1. As discussed in the introduction, the core matching component is the matching and mapping extraction module. The role of matching and mapping extraction module is to discover correct mappings or remove incorrect ones according to specific features extracted from the entities of the input ontologies.

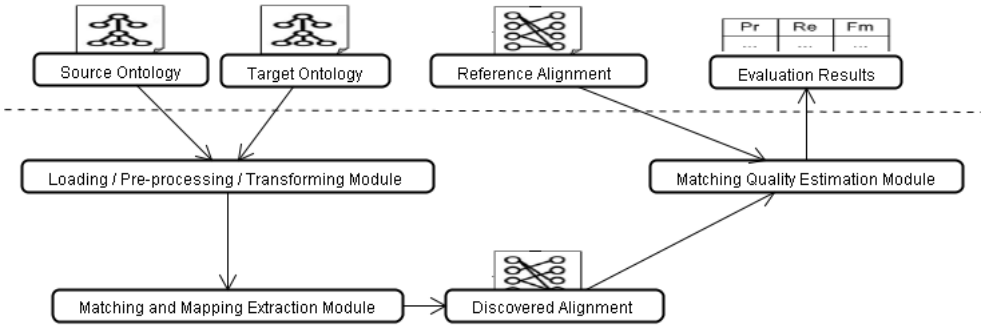

FIGURE 1. Ontology Matching System Architecture

The ontology matching system shown above requires a source and a target ontology and a reference alignment provided by a domain expert as an input. Given a matching scenario, input ontologies are loaded, pre-processed and transformed into internal data structures (Loading/Pre-processing/Transforming module). The Matching Quality Estimation module evaluates the quality of the discovered alignments with the reference alignment. It outputs three evaluation values corresponding to recall, precision and f-measure. In this paper, we compute the harmony means of recall, precision and f-measures on a set of n tests, which are shown as follows:

$$HM(recall) = \frac{\sum_{i=1}^{n}(|A_i \cap R_i|)}{\sum_{i=1}^{n}(|R_i|)} \tag{1}$$

$$HM(precision) = \frac{\sum_{i=1}^{n}(|A_i \cap R_i|)}{\sum_{i=1}^{n}(|A_i|)} \tag{2}$$

$$HM(f-measure) = \frac{2 \times HM(recall) \times HM(precision)}{HM(recall) + HM(precision)} \tag{3}$$

For the ith test, $|A_i|$ denotes the total number of mappings discovered by a matching system, $|A_i \cap R_i|$ refers to the number of correct mappings found, and $|R_i|$ is the number of reference mappings provided by a domain expert. In the sequel, all results will be given by considering f-measures only.

3. Matching and Mapping Extraction.

3.1. **Terminology Based Matcher.** Terminology based matcher discovers mappings by comparing annotations (i.e., labels, comments) of entities. According to [1], among various global terminology based matchers, Information Retrieval Based Mather (IRBM) outperforms the others. IRBM judges the similarity between two entities by the amount of overlap of the information content of their labels [8]. It splits all labels of entities into tokens and calculates the information content of each token in the whole ontology. Then, IRBM extends Tverskys similarity measure [9] with weight of tokens to compute a similarity score between labels of entities. The method compares similarity of two labels by using not only the sequence of characters themselves, but also their information content in an ontology.

3.2. **Structure Based Matcher.** Structure-based matcher discovers mappings between entities by analyzing the similarity of the structural patterns, which these entities are part of. In this paper, we apply the Similarity Propagation (SP) method which is an extension of the well-known similarity flooding algorithm [10]. The basic idea of the method is as follows. Assume that the entities $A1$ and $A2$ in one ontology are related by a directed relation P and the entities $B1$ and $B2$ in another ontology are related by the same directed relation. Then, if we discover that $A1$ and $B1$ is a match, the SP method would imply that $A2$ and $B2$ is a match, too. The similarity values between the two pairs are propagated to each other at each iteration of algorithm. The approach is described in detail in [11].

3.3. **Semantic Based Matcher.** Semantic based matcher is mainly used to refine candidate mappings, which exploits the semantic constraints between entities in the ontologies in order to discover conflicts between potential mappings and remove them from the list of candidate mappings. In this work, we utilize global diagnosis optimization method proposed in [12] which refines the mappings obtained by terminology based matcher in order to remove inconsistent ones.

3.4. **Evolutionary Algorithm for the Matching and Mapping Extraction Problem.** In the matching and mapping extraction module, we need to determine the optimal aggregating parameters and a filter threshold value to select the correct candidate correspondences from the alignments obtained in the matching process. Here, the problem we need to face is how to determine a optimal and robust parameter set so that the final alignment obtained can be of high quality in terms of f-measure. To this end, we construct an optimal model for this problem, which is given as follows:

$$\begin{cases} max \quad f(X) = HM(f - measure(X)) \\ s.t. \quad X = (x_1, x_2, \cdots, x_{n+1})^T \\ \qquad \sum_{i=1}^{n} x_i = 1 \\ \qquad x_i \in [0,1], i = 1, 2, \cdots, n+1 \end{cases} \tag{4}$$

In our work, we take maximizing values of $HM(f - measure)$ as the goal we expect to achieve, and $x_i, i = 1, 2, \cdots, n$ represents the parameter set for aggregating various

TABLE 1. The configurations of Evolutionary Algorithm

Parameter Name	Parameter Value
Numerical accuracy	0.01
Population size	100, the suggested range is [100,300]
Crossover probability	0.7, the suggested range is [0.6, 0.9]
Mutation probability	0.04, the suggested range is [0.01, 0.05]
Max generation	200, the suggested range is [100,300]

matchers, and x_{n+1} for filtering the alignment obtained. We use the binary coding mechanism to express the solution, and the selection, crossover and mutation operators are shown as follows:

3.4.1. *Genetic Operators.* According to the natural law of survival of the fittest, the best individual should have more opportunities of generating offspring. The best chromosomes in a population are the chromosomes that have the best fitness value and the genetic information of these chromosomes can potentially provide the best solutions to the problem. However, reproduction opportunities of the less suitable chromosomes should not be completely removed, because it is important to keep diversity in the population. In this article, 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 evaluation values 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.

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 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.

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

3.4.2. *Algorithm Configuration.* Table 1 shows the parameters of EA which represent a trade-off setting obtained in empirical way to achieve the highest average alignment quality on all test cases of exploited dataset in our experiment:

4. **Experimental Results and Analysis.** In order to study the effectiveness of our proposed scheme, we have exploited a well-known dataset, named benchmark track, provided by the Ontology Alignment Evaluation Initiative (OAEI) 2014 [13] and commonly used for experimentation about ontology alignment problem. In detail, each test case, see Table 2, consists of a set of small scale ontologies which are built around a seed ontology that contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals, and many variations of it. Variations are artificially generated, and focus on the characterization of the behavior of the tools rather than having them compete on real-life problems. They are organized in three groups: Simple tests (1xx) compares the reference ontology with itself; Systematic tests (2xx) are obtained by discarding/modifying features, which include names of entities, comments,

Table 2. Brief Description of OAEI 2014 Benchmark

ID	Brief description
101-104	The ontologies under alignment are the same with each other
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

Table 3. Comparison of our system with the participants in OAEI 2014

Systems	R	P	F
AML	0.39	0.92	0.55
AOT	0.53	0.80	0.64
AOTL	0.53	0.85	0.65
LogMap	0.40	0.40	0.40
LogMap-C	0.40	0.42	0.41
LogMapLite	0.50	0.43	0.46
MaasMatch	0.39	0.97	0.56
OMReasoner	0.50	0.73	0.59
RSDLWB	0.50	0.99	0.66
XMap2	0.40	1.00	0.57
Our System	0.82	0.94	0.88

the specialization hierarchy, instances, properties and classes, from the reference ontology; Real-life ontologies (3xx) are found on the web. In this experiment, we utilize the downloadable datasets from the OAEI 2014 official website for testing purposes.

In order to compare the quality of our proposal with other approaches, we evaluate the obtained alignments with traditional recall, precision and f-measure, and the results in Table 3 are the mean values of all the test cases, and the symbols R, P and F stand for recall, precision and f-measure, respectively. As can be seen from Table 3 that the mean recall and precision of our approach are generally high, and the mean f-measure of the alignments obtained by our approach outperforms all other matching systems. Therefore, through the comparison with the state-of-the-art ontology matching systems, our proposal is effective.

5. **Conclusion.** Due to the high heterogeneity of ontologies, a combination of many methods is necessary to discover correctly the semantic correspondences between their elements. An ontology matching system can be seen as a collection of several matching components, which implements a specific method dealing with a specific heterogeneity type like the terminological, structural and semantic matchers. In this paper, a novel ontology matching system architecture is proposed and the details of its kernel model are presented, i.e. matching and mapping extraction module, where we construct a new optimal model for the ontology matching and mapping extraction problem and design a problem specific EA for it. The experimental results show that the mean recall and precision of our approach is generally high, and the mean f-measure of the alignments obtained by our approach outperform all other state of the art matching systems.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 61503082), Natural Science Foundation of Fujian Province (Nos. 2016J05145 and 2016J05146), Scientific Research Startup Foundation of Fujian University of Technology (No. GY-Z15007), Fujian Province outstanding Young Scientific Researcher Training Project (No. GY-Z160149) and China Scholarship Council.

REFERENCES

[1] D. H. Ngo, Z. Bellahsene, K. Todorov, Opening the black box of ontology matching, Extended Semantic Web Conference, Springer Berlin Heidelberg, pp.16-30, 2013.

[2] H. H. Do and E. Rahm, COMA-a system for flexible combination of schema matching approaches, *Proceedings of the 28th International VLDB Conference*, pp.610-621, 2002.

[3] D. Aumueller, H. H. Do and S. Massmann, Schema and ontology matching with COMA++, *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, pp.906-908, 2005.

[4] C. Drumm, M. Schmitt and H. H. Do, Quickming: automatic schema matching for data migration projects, *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pp.107-116, 2007.

[5] A. Gal, A. Anaby-Tavor and A. Trombetta, A framework for modeling and evaluating automatic semantic reconciliation, *VLDB Journal*, vol.14, no.1, pp.50-67, 2005.

[6] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, USA, 1975.

[7] Martinez-Gil J., Alba E. and Aldana-Montes J. F., Optimizing ontology alignments by using genetic algorithms, *Nature Inspired Reasoning for the Semantic Web (NatuReS2008)*, vol.419, pp.31-45, 2008.

[8] D. H. Ngo, Enhancing Ontology Matching by Using Machine Learning, Graph Matching and Information Retrieval Techniques, PHD thesis, University of Montpellier, 2012.

[9] A. Tversky, Features of similarity, *Psychological Review*, vol.84, no.4, pp.327-352, 1977.

[10] H. G.-M. S. Melnik and E. Rahm, Similarity flooding: A versatile graph matching algorithm and its application to schema matching, *18th International Conference on Data Engineering*, Shanghai, China, April 2002, pp.117-182.

[11] X. Xue, Y. Wang and J. Hou, Ontology Alignment based on Instances using NSGA-II, *Journal of Information Science*, vol.41, no.1, pp.58-70, 2015.

[12] C. Meilicke, Alignment incoherence in ontology matching, PHD thesis, University of Mannheim, 2011.

[13] Ontology Alignment Evaluation Initiative (OAEI) 2014, Available at http://oaei.ontologymatching.org/2014/, Accessed on 15 June, 2016.