Optimizing Ontology Alignment in Vector Space

Shu-Chuan Chu[1], Xingsi Xue[2,3,4,5], Jeng-Shyang Pan[1], Xiaojing Wu[3,4]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] Fujian Key Lab for Automotive Electronics and Electric Drive, Fujian University of Technology, China
[3] College of Information Science and Engineering, Fujian University of Technology, China
[4] Intelligent Information Processing Research Center, Fujian University of Technology, China
[5]Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, China
scchu0803@gmail.com, jack8375@gmail.com, jspan@cc.kuas.edu.tw,
WuXiaojing@smail.fjut.edu.cn

Abstract

Ontology matching technique aims at determining the identical entities, which can effectively solve the ontology heterogeneity problem and implement the collaborations among ontology-based intelligent systems. Typically, an ontology consists of a set of concepts which are described by various properties, and they define a space such that each distinct concept and property represents one dimension in that space. Therefore, it is an effective way to model an ontology in a vector space, and use the vector space based similarity measure to calculate two entities' similarity. In this work, the entities' structure information is utilized to model an ontology in a vector space, and then, their linguistic information is used to reduce the number of dimensions, which can improve the efficiency of the similarity calculation and entity matching process. After that, a discrete optimization model is constructed for the ontology matching problem, and a compact Evolutionary Algorithm (cEA) based ontology matching technique is proposed to efficiently address it. The experiment uses the benchmark track provided by Ontology Alignment Evaluation Initiative (OAEI) to test our proposal's performance, and the comparing results with state-of-the-art ontology matching systems show that our approach can efficiently determine high-quality ontology alignments.

Keywords: Ontology matching, Vector space, Compact evolutionary algorithm

1 Introduction

Nowadays, Artificial Intelligence (AI) has been successfully been applied in various domains [19-25]. As one of them, Semantic Web (SW) attracts more and more attentions. As the kernel technique in SW, an ontology can provide a formal definition on the domain knowledge. However, due to human's subjectivity, the knowledge in different ontologies could be described in different ways. To enable collaborations among ontology-based intelligent systems, it is critical to integrate the knowledge in different ontologies. To this end, ontology matching technique becomes the research hot spot in the SW domain, which dedicates to find the identical entities (e.g., classes and properties) in different ontologies. In general, the procedure of matching ontologies can be divided into two phases: the calculation on the entities' similarity values, and the determination of the identical entity mappings. Typically, an ontology consists of a set of concepts which are described by various properties, which define a space such that each distinct concept and property represents one dimension in that space [3]. Therefore, it is an effective way to model an ontology in a vector space, and use the vector space based similarity measure to calculate two entities' similarity. Tous and Delgado [8] takes the predicates as the references, and on the basis of the relationship between the predicates, each entity is represented as a vector. They further propose a similarity measure based on a matrix representation of all the ontology entities, and update the similarity values inside through a graph matching algorithm. Eidoon et al. [3] make use of the concepts and properties in two ontologies to construct a multi-dimension vector space model, and then vectorize each entity by a weighting mechanism. They calculate two entities' similarity value through the cosine function in the vector space. The existing works mainly model the ontology in the vector space by utilizing the taxonomy information, i.e. the relationships between the entities, whose hereafter matching results could be poor if two ontologies have different structures. To address this issue, we further take into consideration the entities' linguistic information when vectorizing them, which can also reduce the number of vector dimensions and enhance the confidence of the similarity calculation.

Moreover, since the process of determining the identical entity mapping set, i.e. the ontology

[*]Corresponding Author: Xingsi Xue; E-mail: jack8375@gmail.com

DOI: 10.3966/160792642020012101002

alignment, is a complex and time-consuming task, Evolutionary Algorithm (EA) can represent a suitable methodology to find the high-quality alignment [1]. GAOM [9] is the first ontology matching system that utilizes EA to optimize the alignment, which constructs a discrete optimal model for the ontology matching problem. Alves et al. [2] improves GAOM by introducing the local search strategy into the algorithm's evolving process, and they also take into consideration the information of the instances when calculating two entities' similarity value. More recently, Xue and Wang [14] propose a Memetic Algorithm (MA), a hybrid EA, to solve the ontology matching problem in the Linked Open Data cloud (LOD). The existing EA-based matchers are implemented on the population evolving mechanism, which requires huge memory consumption. To overcome this drawback, in this work, a novel optimal model for the ontology matching problem is constructed, and then, a compact EA (cEA), which replaces the population with a probability representation, is proposed to efficiently determine the high-quality alignment.

The rest of the paper is organized as follows: Section 2 defines the ontology and ontology matching problem; Section 3 present the method that models an ontology modeling in the vector space; Section 4 describe in details the cEA-based ontology matching technique; Section 5 presents the experimental studies and analysis; finally, Section 6 draws a conclusion.

2 Ontology and Ontology Matching Problem

An ontology is defined as 3-tuple (C, P_d, P_o), where C, P_d, and P_o are respectively referred to the set of concepts, datatype properties and object properties. In general, C, P_d, and P_o are called ontology entities. Since different ontologies are developed and maintained by different domain experts, the knowledge in them could be described in different ways. To bridge the semantic gap between two ontologies, we need to determine an ontology alignment. In particular, an ontology alignment is an entity mapping set, where each mapping is a 4-tuple (e, e', =, *simValue*) where e and e' are respectively the entities from two ontology, = is the relation of equivalence between e and e', and s*imValue* is the similarity value between e and e'.

A similarity measure takes as input two ontology entities' information, and outputs a real number in [0, 1] to show to what extent they are similar. In particular, 1 means they are identical, and 0 means the opposite. Similarity measure is the foundation of an ontology matching technique, which directly affects the quality of an ontology alignment. In this work, we utilize the cosine similarity measure to calculate two entities' similarity value, which is defined as follows:

$$cosine\left(\overrightarrow{x_1}, \overrightarrow{x_2}\right) = \frac{\overrightarrow{x_2} \cdot \overrightarrow{x_1}}{\left\|\overrightarrow{x_1}\right\| \cdot \left\|\overrightarrow{x_2}\right\|} \tag{1}$$

where $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$ respectively corresponds to two entities in the ontologies. Before calculating two entities' cosine similarity value, they should be respectively represented in the vector space. With respect to the ontology representation in the vector space, please see also Section 3.

It is obvious that, if the golden standard alignment is 1:1, the quality of an alignment is proportional to the cardinality of it and the mean similarity value of all entity correspondences inside. On this basis, we propose the following metrics to approximately measure the quality of an ontology alignment:

$$r(A) = \frac{|A|}{min\{|O_1|, |O_2|\}} \tag{2}$$

$$p(A) = \frac{\sum simValue_i}{|A|} \tag{3}$$

$$f(A) = \sqrt{r(A) \cdot p(A)} \tag{4}$$

where O_1 and O_2 are two ontologies under alignment, and A is an alignment between them; $|O_1|$, $|O_2|$ and $|A|$ are respectively the number of entities inside O_1 and O_2, and the amount of mappings inside A; $simValue_i$ is the i-th entity mapping's similarity measure, which is defined in Eq. 1. Particularly, $r(x)$ approximately measures the fraction that the amount of correct mappings found divided by the number of all correct mappings, $r(x)$ approximately calculates the fraction that the quantity of correct mappings found divided by the number of all found mappings, and the $f(x)$ is the mean value of them.

$$\begin{aligned} & max \quad f(X) \\ & s.t. \quad X = (x_1, x_2, \dots, x_{|O_1|})^T \\ & x_i \in \{1, 2, ..., |O_2|\}, i = 1, 2, ..., |O_1| \end{aligned} \tag{5}$$

where $|O_1|$ and $|O_2|$ are respectively the cardinalities of two ontologies O_1 and O_2, x_i, i = 1, 2, …, O_1 is the i-th correspondence, a decision variable X in the solution space corresponds to an ontology alignment in the problem space, and the objective function is to maximize the alignment's f().

3 Ontology Vectorization

3.1 The Dimension of Vector Space

Model an ontology in a vector space is to make a vector space that any of its dimensions represents a concept or a property of two ontologies. In particular, dissimilar concepts or properties will not be duplicated in the vector space, and the vector space should fully cover all the concepts and properties that exist in two ontologies. In this work, the vector space is built by extracting all concepts, datatype properties and object properties from two ontologies as the dimensions. Then each entity is presented as a vector in this vector space with a weighting mechanism.

Given a source ontology and a target ontology, the dimensions of vector space are built as follow:

$$\{C_{src} - C_{anchor}, C_{anchor}, C_{tgt} - C_{anchor}, P_{d,\ src} - P_{d,\ anchor}, P_{d,\ anchor}, P_{d,\ tgt} - P_{d,\ anchor}, P_{o,\ src} - P_{o,\ anchor}, P_{o,\ anchor}, P_{o,\ tgt} - \mathrm{P}_{p,\ anchor}\},$$

where C_{src}, C_{tgt}, $P_{d,src}$ $P_{d,tgt}$, $P_{o,src}$ and $P_{o,tgt}$ are respectively two ontologies' concept set, datatype property set and object property set; C_{anchor}, $P_{d,anchor}$ and $P_{o,anchor}$ are respectively the identical concepts, datatype properties and object properties.

The determination of the anchors of concepts and properties can be of help to reduce the vector space's dimension number and introduce the linguistic information of the entities when vectorizing them, which can reduce the computation complexity on the entity similarity calculation and improve the robustness of the matcher especially when two ontologies has different structures. In this work, the hash tables are utilized to determine the entity anchors by using their labels or comments, then a weighting strategy is used to vectorize each entity.

3.2 Entity Weighting Approach

Concept weighting. In this section, we present the weighting formulas on a source concept, and a target entity's weights can be calculated in a similar way.

Let c be a source concept dimension or anchor concept dimension, a source concept x's concept weight in this dimension is calculated as follows:

$$w_c(x) = \begin{cases} \dfrac{dist(x,c)}{dist(x,ROOT)}, \text{if } x \text{ is } c's \text{ sub - concept} \\ \dfrac{dist(x,c)}{dist(x,ROOT)}, \text{if } x \text{ is } c's \text{ supper - concept} \\ 1, \text{if } dust(x,c) \\ 0, \text{otherwise} \end{cases} \quad \textbf{(6)}$$

where $dist(x, c)$ is the shortest path's length between x and c in the concept hierarchy graph, and $dist(x, root)$ (or $dist(c, root)$) is the shortest path's length between x (or c) and $ROOT$ that is the root node of the concept hierarchy graph.

Let p_d be a source datatype property dimension or anchor datatype property dimension, a source concept x's datatype property weight in this dimension is calculate as follows:

$$w_{p_d}(x) = \begin{cases} 1, \text{if } x \text{ has } p_d \\ 0, \text{otherwise} \end{cases} \quad \textbf{(7)}$$

Let p_o be a source object property dimension or anchor object property dimension, a source concept x's object property weight in this dimension is calculate as follows:

$$w_{p_o}(x) = \begin{cases} 1, \text{if } x \text{ is } p_o^{'}s \text{ domain or range} \\ 0, \text{otherwise} \end{cases} \quad \textbf{(8)}$$

For the target concept dimensions, the target datatype property dimensions and target object property dimensions, x's concept weights in the corresponding dimension are 0.

Datatype property. Weighting Let c be a concept dimension, a datatype property x's concept weight in this dimension is calculated as follows:

$$w_c(x) = \begin{cases} 1, \text{if } c \text{ has } x \\ 0, \text{otherwise} \end{cases} \quad \textbf{(9)}$$

Let p_d be a datatype property dimension, a datatype property x's datatype property weight in this dimension is calculate as follows:

$$w_{p_d}(x) = \begin{cases} 1, \text{if } x \text{ and } p_d \text{ have the same name} \\ 0, \text{otherwise} \end{cases} \quad \textbf{(10)}$$

For the object property dimensions, x's object property weights are 0.

Object property weighting. Let c be a concept dimension, an object property x's concept weight in this dimension is calculated as follows:

$$w_c(x) = \begin{cases} 1, \text{if } c \text{ is } x's \text{ domain or range} \\ 0, \text{otherwise} \end{cases} \quad \textbf{(11)}$$

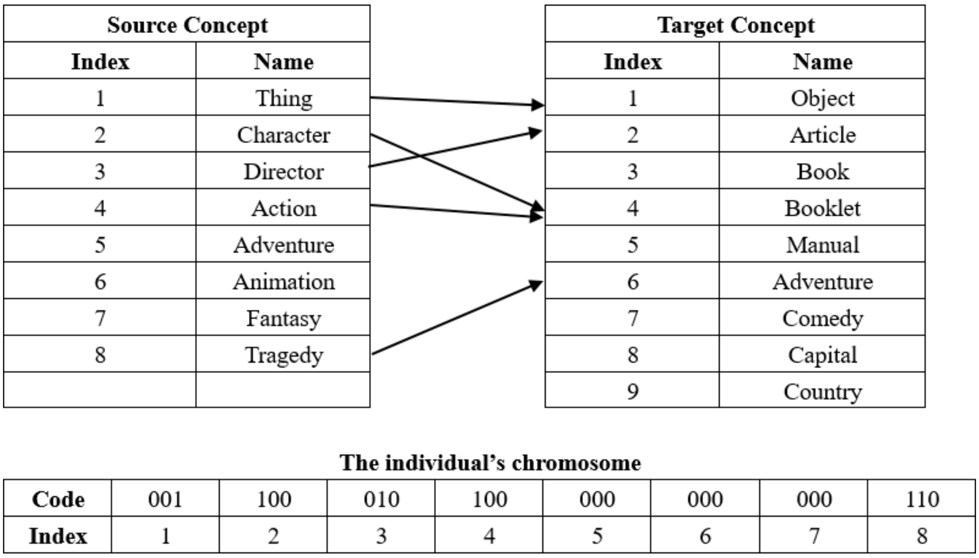
Let p_o be an object property dimension, an object property x's object property weight in this dimension is calculate as follows:

$$w_{p_o}(x) = \begin{cases} 1, \text{if } x \text{ is } p_o \text{ have the same name} \\ 0, \text{otherwise} \end{cases} \quad \textbf{(12)}$$

For the datatype property dimensions, x's datatype property weights are 0.

4 Compact Evolutionary Algorithm Based Ontology Matching Technique

Since matching ontologies is a complex and time-consuming task, EA becomes a suitable method of addressing it [13-18]. In this paper, we further propose a compact EA-based ontology matching technique, which replaces the explicit representation of the population with a probability distribution to save the memory consumption. Here, we empirically choose the Gray code [5], which is a binary encoding mechanism, to encode an alignment to ensure cEA's evolving efficiency. An example of the encoding mechanism is shown in the Figure 1, in particular, Gray code 000 means a source concept is not mapped to any target concept.



Source Concept	
Index	**Name**
1	Thing
2	Character
3	Director
4	Action
5	Adventure
6	Animation
7	Fantasy
8	Tragedy

Target Concept	
Index	**Name**
1	Object
2	Article
3	Book
4	Booklet
5	Manual
6	Adventure
7	Comedy
8	Capital
9	Country

The individual's chromosome

Code	001	100	010	100	000	000	000	110
Index	1	2	3	4	5	6	7	8

Figure 1. An example of encoding and decoding mechanism

cEA uses a Probability Vector (PV) [12] to characterize the population. A PV's dimension is equal to the number of a solution's gene bit, and each dimension's range is [0, 1]. In particular, the value in each PV's dimension represents a probability of being 1 on a solution's corresponding gene bit. Therefore, we can utilize a PV to generate different solutions. For example given PV= (0.1, 0.3, 0.5) ^{T}, we first generate three random real numbers in [0, 1], say 0.6, 0.2 and 0.4. Since 0.6 > 0.1, 0.2 < 0.3 and 0.4 < 0.5, the newly generated solution is 100. In each generation, we will update PV by moving it to the elite. In particular, given an update step *st*, if the gene value of the elite solution is 1, the corresponding dimension number of PV will increase by *st*, otherwise decrease by *st*.

Given the maximum generation *MaxGeneration* = 2000, the step length for updating PV *st* = 0.1, the pseudocode of cEA is given in Algorithm 1. In each generation, cEA generates a new solution through PV, and tries to update PV through the competition between it and elite solution. When all the elements in PV is 1 or 0, cEA converges.

Algorithm 1. Compact Evolutionary Algorithm

```
**** Initialization ****
for $i = 0$; $i < PV.length$; $i$++ do
    $PV_i = 0.5$;
end for
$solution_{elite}$=generate a solution through PV;
generation=1;
**** Evolving Process ****
while $generation < MaxGeneration$ do
    $solution_{new}$=generate a solution through PV;
    $[winner, loser] = compete(solution_{new}, solution_{elite})$;
    if $winner == solution_{new}$ then
        $solution_{elite} = solution_{new}$;
    end if
    for $i = 0$; $i < PV.length$; $i$++ do
        if $winner_i == 1$ then
            $PV_i = PV_i + st$;
        else
            $PV_i = PV_i - st$;
        end if
    end for
    $generation = generation + 1$;
end while
```

5 Experimental Studies and Analysis

Testing Cases and Experimental Configuration

In the experiments, the benchmark track provided by the Ontology Alignment Evaluation Initiative (OAEI) are used to test our approach's performance. Each testing case in the benchmark track consists of two ontologies under alignment (a source ontology which is a seed ontology, and a target ontology which is a variance of the seed ontology) and a reference alignment for evaluating the ontology matcher's effectiveness. We compare cEA with VBOM [3] and GAOM which are respectively the state-of-the-art vector space based matcher and the EA-based matcher [9]. CEA uses the parameters (see also Section 4) which represent a trade-off setting obtained in an empirical way to achieve the highest average alignment quality, VBOM and GAOM's configurations are referring to their own literatures. The results of cEA and GAOM shown in the tables are the mean values of thirty independent executions. In order to compare with different matching techniques, in this work, we use the recall, precision and f-measure [7] to evaluate an alignment's quality. In particular, given a reference alignmentA_{ref}, the recall, precision and f-measure of an alignment A are defined as follows:

$$recall(A)=\frac{|A\cap A_{ref}|}{|A_{ref}|} \quad \textbf{(13)}$$

$$precision(A)=\frac{|A\cap A_{ref}|}{|A|} \quad \textbf{(14)}$$

$$f-measure(A)=\frac{2\cdot recall(A)\cdot precision(A)}{recall(A)+precision(A)} \quad \textbf{(15)}$$

Essentially, precision and recall are the ratio of the number of true positive ($|$ A $\cap$ A_{ref} $|$) on that of the retrieved correspondences ($|$ A $|$) and those expected (A_{ref}), respectively, and f-measure is their harmony mean [4].

Table 1 and Table 2 compare our approach with VBOM and GAOM by carrying out the T-test statistical analysis [6] on the alignment's quality. All the competitors are run for 30 independent executions on each testing case, and thus, we need consider the critical value for 29 degrees of freedom, which is equal to 2.045.

Table 1. Comparison of the alignments in terms of f-measure and standard deviation *stDev*

ID	VBOM *f-measure* (*stDev*)	GAOM *f-measure* (*stDev*)	Our Approach *f-measure* (*stDev*)	Descriptions
Task1: Ontologies with similar labels				
101	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	Reference
103	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	Language Generalization
104	1.00 (0.00)	0.98 (0.02)	1.00 (0.00)	Language restriction
221	1.00 (0.00)	0.99 (0.01)	1.00 (0.01)	No specialization
222	0.90 (0.02)	0.98 (0.02)	1.00 (0.01)	Flattened hierarchy
223	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	Expanded hierarchy
224	1.00 (0.00)	0.99 (0.01)	1.00 (0.02)	No instance
225	1.00 (0.00)	0.99 (0.01)	1.00 (0.02)	No restrictions
228	1.00 (0.00)	0.98 (0.02)	1.00 (0.02)	No properties
230	1.00 (0.00)	1.00 (0.01)	1.00 (0.01)	Flattened classes
231	1.00 (0.00)	0.98 (0.01)	1.00 (0.01)	Expanded classes
Task2: Ontologies with similar structures and different labels				
201	0.91 (0.02)	0.93 (0.02)	0.94 (0.01)	No names
202	0.91 (0.02)	0.96 (0.01)	0.96 (0.01)	No names, No comments
203	1.00 (0.00)	1.00 (0.01)	1.00 (0.00)	No comments
204	0.95 (0.01)	1.00 (0.00)	1.00 (0.00)	Naming conventions
205	0.91 (0.01)	0.93 (0.02)	0.95 (0.02)	Synonyms
206	0.81 (0.02)	0.80 (0.02)	0.88 (0.02)	Translation
Task3: Real Ontologies with different labels and structures				
301	0.61 (0.01)	0.59 (0.01)	0.68 (0.01)	Real: BibTeX/MIT
302	0.59 (0.02)	0.56 (0.03)	0.63 (0.02)	Real: BibTeX/UMBC
303	0.51 (0.01)	0.48 (0.03)	0.67 (0.02)	Real: Karlsruhe
304	0.86 (0.01)	0.84 (0.02)	0.88 (0.01)	Real: INRIA

Table 2. T-Test statistical analysis on the alignment's quality

ID	(Our Approach, VBOM) *t-value*	(Our Approach, GAOM) *t-value*
Task1: Ontologies with similar labels		
101	0.00	0.00
103	0.00	0.00
104	0.00	5.47
221	0.00	3.87
222	24.49	7.74
223	0.00	0.00
224	0.00	2.44
225	0.00	2.44
228	0.00	3.87
230	0.00	0.00
231	0.00	7.74

Table 2. (continue)

ID	(Our Approach, VBOM) *t-value*	(Our Approach, GAOM) *t-value*
Task2: Ontologies with similar structures and different labels		
201	7.34	2.44
202	12.24	0.00
203	0.00	0.00
204	27.38	0.00
205	4.89	0.00
206	7.74	9.68
Task3: Real Ontologies with different labels and structures		
301	15.49	23.23
302	7.74	10.63
303	7.34	9.11
304	7.74	9.79

As can be seen from Table 2, the alignments obtained by cEA are all better than or equal to VBOM and GAOM in all three categories of testing cases. Particularly, VBOM does not perform well when matching ontologies without label and structure information in task3, GAOM is not stable especially in task1, and all three competitors perform well in task2 whose ontologies have similar structures. Our approach models an ontology in a vector space, which allows it take full advantage of the ontology entity's linguistic and structure information to distinguish identical entities, and models the ontology matching process as a discrete optimization problem, which can ensure the alignment's quality when the ontologies under alignment lack of label and structure information. In addition, VBOM and GAOM respectively requires 72,602 and 156,480 bytes in average for matching each testing cases, while cEA's mean memory consumption on each testing case is only 38,610 bytes, which improves VBOM and GAOM by 46.81% and 75.32%, respectively.

In Figure 2, X-axis is different ontology matchers and Y-axis is the f-measure value. As can be seen from Figure 2, our approach's f-measure is 0.9, which outperforms all the OAEI's participant in terms of f-measure. Since evolutionary approaches are efficient in finding an isomorphism between the subgraphs modeling the two ontologies, particularly when the considered ontologies are characterized by a significant number of entities [10-11]. The quality of the alignments obtained by our approach is in general better than those by the state-of-the-art ontology matching systems without using evolutionary approach. To conclude, our approach can efficiently determine high-quality ontology alignments when matching various heterogeneous ontologies.

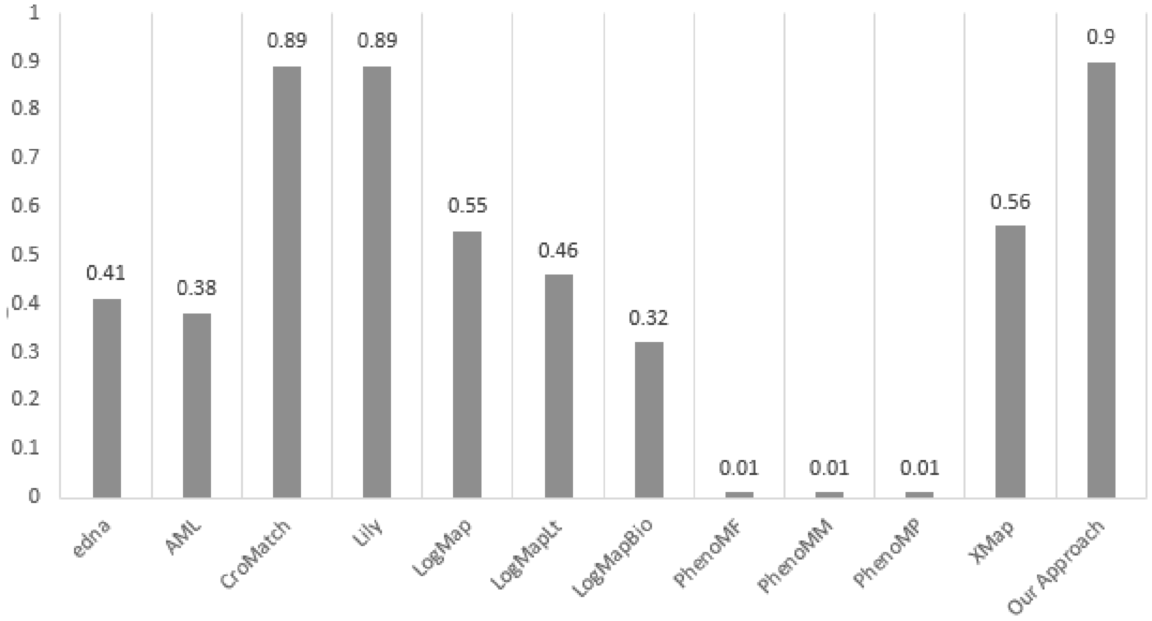


Figure 2. Comparison among our approach and OAEI's participants in terms of f-measure

6 Conclusion

Ontology matching aims at finding identical entities in different ontologies, which can bridge the semantic gap between heterogeneous ontologies and integrate the knowledge inside. To efficiently matching ontologies, in this work, we first model an ontology in a vector space with entities' linguistic and structure information, and then utilize two vector's cosine distance to measure the similarity of two corresponding entities. On this basis, we further model ontology matching process as a discrete optimization problem, and propose a compact EA to address it. The experiment utilizes OAEI's benchmark track to test our approach's performance, and the comparing results with VBOM, GAOM and OAEI's participants show that our approach can determine high-quality ontology alignments with low memory consumptions.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61503082), the Natural Science Foundation of Fujian Province (Nos. 2016J05145 and 2017J01514), the Program for New Century Excellent Talents in Fujian Province University (No. GY-Z18155), the Program for Outstanding Young Scientific Researcher in Fujian Province University (No. GY-Z160149), the Scientific Research Foundation of Fujian University of Technology (Nos. GY-Z17162 and GY-Z15007) and the Humanities and Social Sciences Foundation of Ministry of Education of China (No. 17YJC630213).

References

[1] G. Acampora, V. Loia, A. Vitiello, Enhancing Ontology Alignment through a Memetic Aggregation of Similarity Measures, *Information Sciences*, Vol. 250, pp. 1-20, November, 2013.

[2] A. Alves, K. Revoredo, F. Baiao, Ontology Alignment Based on Instances Using Hybrid Genetic Algorithm, The *Proceedings of 7th International Conference on Ontology Matching*, Boston, MA, USA, 2012, pp. 242-243.

[3] Z. Eidoon, N. Yazdani, F. Oroumchian, Ontology Matching Using Vector Space, *European Conference on Information Retrieval*, Glasgow, UK, 2008, pp. 472-481.

[4] J. Euzenat, Semantic Precision and Recall for Ontology Alignment Evaluation, *International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007, pp. 348-353.

[5] G. Frank, *Pulse Code Communication*, U.S. Patent 2,632,058, March, 1953.

[6] J.-S. Milton, J. C. Arnold, *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*, McGraw-Hill Publishing Company, 1990.

[7] C.-J. Van Rijsbergen, Foundation of Evaluation, *Journal of Documentation*, Vol. 30, No. 4, pp. 365-373, April, 1974.

[8] R. Tous, J. Delgado, A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment, *International Conference on Database and Expert Systems Applications*, Kraków, Poland, 2006, pp. 307-316.

[9] J. Wang, Z. Ding, C. Jiang, GAOM: Genetic Algorithm Based Ontology Matching, *2006 IEEE Asia-Pacific Conference on Services Computing*, Guangzhou, China, 2006, pp. 617-620.

[10] X. Xue, J. Chen, Optimizing Ontology Alignment through Hybrid Population-Based Incremental Learning Algorithm, *Memetic Computing*, Vol. 11, No. 2, pp. 209-217, June, 2019.

[11] X. Xue, J. Chen, X. Yao, Efficient User Involvement in Semiautomatic Ontology Matching, *IEEE Transactions on Emerging Topics in Computational Intelligence*, December, 2018, DOI: 10.1109/TETCI.2018.2883109.

[12] X. Xue, J.-S. Pan, A Compact Co-Evolutionary Algorithm for Sensor Ontology Meta-Matching, *Knowledge and Information Systems*, Vol. 56, No. 2, pp. 335-353, August, 2018.

[13] X. Xue, Y. Wang, Optimizing Ontology Alignments through a Memetic Algorithm Using both MatchFmeasure and Unanimous Improvement Ratio, *Artificial Intelligence*, Vol. 223, pp. 65-81, June, 2015.

[14] X. Xue, Y. Wang, Using Memetic Algorithm for Instance Coreference Resolution, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 2, pp. 580-591, February, 2016.

[15] Z. Meng, J.-S. Pan, K.-K. Tseng, PaDE: An Enhanced Differential Evolution Algorithm with Novel Control Parameter Adaptation Schemes for Numerical Optimization, *Knowledge-Based Systems*, Vol. 168, pp. 80-99, March, 2019.

[16] Z. Meng, J.-S. Pan, HARD-DE: Hierarchical ARchive Based Mutation Strategy with Depth Information of Evolution for the Enhancement of Differential Evolution on Numerical Optimization, *IEEE Access*, Vol. 7, pp. 12832-12854, January, 2019.

[17] Z. Meng, J.-S. Pan, Monkey King Evolution: A New Memetic Evolutionary Algorithm and Its Application in Vehicle Fuel Consumption Optimization, *Knowledge-Based Systems*, Vol. 97, pp. 144-157, April, 2016.

[18] Z. Meng, J.-S. Pan, L. Kong, Parameters with Adaptive Learning Mechanism (PALM) for the Enhancement of Differential Evolution, *Knowledge-Based Systems*, Vol. 141, pp. 92-112, February, 2018.

[19] J.-S. Pan, P. Hu, S.-C. Chu, Novel Parallel Heterogeneous Meta-Heuristic and Its Communication Strategies for the Prediction of Wind Power, *Processes*, Vol. 7, No. 11, pp. 845-869, November, 2019.

[20] J.-S. Pan, C.-Y. Lee, Sghaier A, Zeghid M, Xie J. Novel Systolization of Subquadratic Space Complexity Multipliers Based on Toeplitz Matrix-Vector Product Approach, *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 27, No. 7, pp. 1614-1622, July, 2019.

[21] T.-T. Nguyen, J.-S. Pan, T.-K. Dao, An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network, *IEEE Access*, Vol. 7, pp. 75985-75998, June, 2019.

[22] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, V. Snasel , A Clustering Scheme for Wireless Sensor Networks Based on Genetic Algorithm and Dominating Set, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 1111-1118, July, 2018.

[23] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, G. J. Kim, A PSO Based Energy Efficient Coverage Control Algorithm for Wireless Sensor Networks, *Computers, Materials & Continua*, Vol. 56, No. 3, pp. 433-446, November, 2018.

[24] J. Wang, Y. Gao, W. Liu, W. Wu, S.-J. Lim, An Asynchronous Clustering and Mobile Data Gathering Schema based on Timer Mechanism in Wireless Sensor Networks, *Computers, Materials & Continua*, Vol. 58, No. 3, pp. 711-725, January, 2019.

[25] C. Yin, S. Ding, J. Wang, Mobile Marketing Recommendation Method Based on User Location Feedback, *Human-centric Computing and Information Sciences*, Vol. 9, No. 1, pp. 1-17, December, 2019.

Biographies

Shu-Chuan Chu received the Ph.D. degree in School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She served as an Assistant Professor and Associate Professor of Department of Information Management, Cheng Shiu University, Taiwan. Currently, she is a Research Fellow in the School of Computer Science, Engineering and Mathematics of Flinders University, Australia. Her research interests are mainly in Data Mining, Computational Intelligence, Information Hiding and Signal Processing.

Xingsi Xue received the B.S. degree in Software Engineering from Fuzhou University, China in 2004, the M. S. degree in Computer Application Technology from Renmin University of China, China in 2009, and the Ph.D. degree in Computer Application Technology from Xidian University, China in 2014. He is a professor at the College of Information Science and Engineering, Fujian University of Technology. His research interests include intelligent computation, data mining and large-scale ontology matching technology.

Jeng-Shyang Pan received the B.S. degree in Electronic Engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in Communication Engineering from the National Chiao Tung University, Taiwan in 1988, and the Ph.D. degree in Electrical Engineering from the University of Edinburgh, U.K. in 1996. Currently, he is the Assistant President in Fujian University of Technology and the Doctoral Advisor in the Harbin Institute of Technology Shenzhen Graduate School. His current research interests include soft computing, robot vision and signal processing.

Xiaojing Wu received the bachelor’s degree from Fuzhou University, Fujian, China, in 2018. Now, she is a graduate student at the college of information and science engineering, Fujian University of Technology. Her research domain include Intelligent Computation, Ontology Visualization and Ontology Matching Technique.