Hindawi
Mobile Information Systems
Volume 2022, Article ID 8587896, 7 pages
https://doi.org/10.1155/2022/8587896

Research Article

Integrating Cross-lingual Ontologies through Co-Evolutionary Algorithm

Lili Huang[1] **and Leong Ko**[2]

[1]*International College Fujian Agriculture and Forestry University, No. 15 Shangxiadian Road, Changshan District, Fuzhou, Fujian, 350002, China*
[2]*School of Languages and Cultures University of Queensland Bldg. 32 Gordon Greenwood Building, The University of QLD, Brisbane, Queensland, 4072, Australia*

Correspondence should be addressed to Lili Huang; lilihuang@vip.163.com

Received 13 January 2022; Accepted 25 February 2022; Published 5 April 2022

Academic Editor: Jianhui Lv

Copyright © 2022 Lili Huang and Leong Ko. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To support the collaborations among intelligent applications, it is necessary to integrate various ontologies which are developed and maintained by different organizations. One of the challenges is that different ontologies in the same application domain might use different languages to describe the same concept, which yields the cross-lingual heterogeneity problem, i.e., how to map two identical entities in different languages. To address this problem, this work proposes a problem-specific co-evolutionary algorithm (CoEA)-based matching technique. In particular, we first propose a parallel aggregating framework to aggregate different SMs and then construct a continuous optimization model for defining the problem of cross-lingual ontology integration. To better trade off the algorithm's exploitation and exploration, we use two competitive subpopulations to, respectively, execute the exploitation and exploration. The experiment utilizes OAEI's multifarm track for testing purpose, and the experimental results show that CoEA is able to effectively integrate various cross-lingual ontologies.

1. Introduction

With the rapid development of Semantic Web [1, 2], various ontologies have been developed in diverse domains to annotate the data. To implement the intelligent applications' collaboration, it is necessary to integrate various ontologies which are developed and maintained by different organizations. One of the challenges is that different ontologies in the same application domain might use different languages to describe the same concept, which yields the cross-lingual heterogeneity problem. To address this issue, we need to map two identical entities in different languages, which is the so-called cross-lingual ontology matching [3]. When matching two ontologies, it is important to use the similarity measure (SM) to distinguish the heterogeneous entities [4], and usually, different SMs should be aggregated to make their advantages and disadvantages complement each other, which is of help to improve the confidence of the result. Currently, the parallel framework owns such merits as its flexibility of tuning various weights and the relative independence on different matchers in terms of their executing processes. Before aggregating SMs, their corresponding similarity matrices should be calculated independently first. The similarity matrix's row and column are, respectively, two ontology's entities, and its elements are the corresponding entities' similarity value. Then, these matrices are aggregated into one matrix by using the aggregating weights. Finally, a threshold is used to filter the correspondences with low similarity values.

Since it is a complex optimizing task of optimizing a cross-lingual ontology alignment, metaheuristics approaches, such as evolutionary algorithm (EA) [5],

become the popular methods of determining the high-quality alignments. Being inspired by the success of metaheuristics approaches in the cross-lingual ontology matching domain, this work proposes a co-evolutionary algorithm (CoEA) to determine the cross-lingual ontology alignment. CoEA uses the compact mechanism and co-evolutionary mechanism to overcome two shortcomings of classic EA, i.e., high computational cost and premature convergence. In particular, we make the following contributions: a parallel aggregating framework of matching cross-lingual ontologies is presented; a continuous optimization model is built to define the cross-lingual ontology matching problem; a problem-specific CoEA is proposed to optimize the cross-lingual ontology alignment's quality, which uses two competitive subpopulations to trade off algorithm's exploitation and exploration.

The rest of this study is organized as follows. After defining the cross-lingual ontology matching problem (Section 2), three kinds of SMs are introduced (Section 3), then CoEA is presented (Section 4), and the experimental results are shown (Section 5). Finally, the conclusion is drawn (Section 6).

2. Cross-Lingual Ontology Integrating Problem

An ontology consists of concepts, properties, and axioms, and an ontology alignment is a mapping set between two heterogeneous ontologies. A mapping is a 3-tuple (c_1, c_2, $simValue$), where c_1 and c_2 are, respectively, two ontologies' entities, and $simValue \in\in [0, 1]$ is their similarity [6, 7]. Given an alignment A and a reference alignment A_{ref}, A's quality can be measured with f-measure [8]:

$$\begin{aligned} \text{recall}(A) &= \frac{\left|A \cap A_{ref}\right|}{\left|A_{ref}\right|}, \\ \text{precision}(A) &= \frac{\left|A \cap A_{ref}\right|}{|A|}, \\ f-\text{measure}(A) &= \frac{2 \times \text{recall}(A) \times \text{precision}(A)}{\text{recall}(A) + \text{precision}(A)}, \end{aligned} \tag{1}$$

where $|\,|$ is the cardinality of a particular set. The cross-lingual ontology integrating problem is modelled as a continuous optimization problem. In particular, its objective is to maximize the f-measure of the cross-lingual ontology alignment, and its decision variable $= \{x_1, x_2 \ldots\}^T$, where $x_i \in [0, 1]$ is the ith aggregating weight, and their sum is equal to 1.

3. Similarity Measure

Generally, there are three broad categories of SM, i.e., syntactic SM, linguistic SM, and taxonomy SM [9–11]. Syntactic SM calculates two strings' similarity by measuring their edit distance. Levenshtein distance [12] is one of the popular syntactic SMs, which is defined as follows:

$$sim_{\text{Levenshtein}}(s_1, s_2) = \max\left(0, \frac{\min\left(\left|s_1\right|, \left|s_2\right|\right) - \text{d}(s_1, s_2)}{\min\left(\left|s_1\right|, \left|s_2\right|\right)}\right), \tag{2}$$

where is the character numbers of a string and $d()$ is two strings' edit distance.

Linguistic SM measures two words' similarity with the electronic dictionary, such as Wordnet [13,14]. Given two entities' label label_1 and label_2, their linguistic similarity value is defined as follows:

$$sim_{\text{Levenshtein}}(\text{label}_1, \text{label}_2) = \begin{cases} 1, & \text{if } \text{label}_1 \text{ and } \text{label}_2 \text{ are synonymous,} \\ 0.5, & \text{if } \text{label}_1 \text{ and } \text{label}_2 \text{ are hyponymous or hypernymous,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Taxonomy SM uses two concepts c_1 and c_2's context to determine their similarity, which is defined as follows [15, 16]:

$$sim_{SF}(c_1, c_2) = \frac{sim_{\text{Levenshtein}}(\text{super}_1, \text{super}_2) + \sum sim_{\text{Levenshtein}}\left(sub_i, sub_j\right)}{2}, \tag{4}$$

where super_1 and super_2 are, respectively, c_1 and c_2's super classes and sub_i and sub_j are, respectively, their ith and jth subclasses.

In this study, by referring to Chen et al.'s work [17], we first use the Babelnet Translate [1] to translate the terminologies in various languages into English and then use the above three SMs to calculate their similarity values. All the similarity measures are executed in parallel to determine their corresponding similarity matrices, which store the similarity values of all correspondences. After that, CoEA is used to determine the aggregating weight for the similarity matrices in this parallel framework.

4. Co-Evolutionary Algorithm

4.1. Compact Encoding. CoEA uses the binary coding mechanism [18], which is of help to reduce the algorithm's computational complexity. Given a set of cut points $C^{'} = \{c_1, c_2, \ldots, c_n\}$, we first sort it in ascending order as $C = \{c_1, c_2, \ldots, c_n\}$, and then, we can get the corresponding weight set through the following equation:

$$w_k = \begin{cases} c_1, & k = 1 \\ c_k - c_{k-1}, & 1 < k < p \\ 1 - c_{p-1}, & k = p \end{cases}. \quad (5)$$

After that, we use n cutting points to obtain $n+1$ aggregating weights.

This work uses two probability vectors (PVs) to describe the gene distribution on two competitive subpopulations, i.e., PV_{better} and PV_{worse}. In particular, PV_{better} describes the subpopulation whose elite solution owns higher fitness value, while PV_{worse} is lower. Each PV's elements are the real number in [0,1], which represents their corresponding gene bit's probability of being 1. With a PV, we are able to generate various solutions with the similar gene distributions.

4.2. Exploration and Exploitation. We apply different strategies on PV_{better} and PV_{worse}'s corresponding subpopulations. The former mainly focuses on the exploitation operator, while the latter uses the exploration strategy. For the sake of clarity, we show these two strategies in Algorithms 1 and 2.

Here, we introduce the exponential crossover operator (EC) [19] to implement CoEA's exploration and exploitation operators. Comparing with traditional crossover operator, EC generates the offspring by inheriting a complete sequential genes from two parents, which is more exploitative. Given two solutions, EC randomly copies a certain number of sequential bits' values from the first one to the second one. With respect to the exploration operator, we use EC to mix a newly generated solution $\text{solution}^{\text{new}}$ and the elite pollen $\text{solution}^{\text{elite}_{\text{better}}}$, while in the exploitation operator, we first mix two newly generated solutions $solution^p$ and $solution^q$ to obtain the mediate individual; then, we mix it with $solution^{new}$, which approximates the evolutionary operator of differential evolution algorithm (DE) [20] to ensure the algorithm's exploration.

```
(1) solution^new = PV^better. generateSolution();
(2) int num = round(ran(0, 1) × solution.length);
(3)  int index = 0;
(4) for int i = 0; i < solution^new.length; i++ do
(5)    if index + 1 > num then
(6)       break;
(7)    end if
(8) if index + 1 > solution.length then
(9)       index = 0;
(10)   end if
(11)   solution_i^new = solution_i^elite_better;
(12) end for
```

Algorithm 1: Exploration.

```
(1) solution^new = PV^worse.generateSolution();
(2) solution^p = PV^worse.generateSolution();
(3) solution^q = PV^worse.generateSolution();
(4) int num = round(ran(0, 1) × solution.length);
(5) int index = 0;
(6)    for int i = 0; i < solution^p.length; i++ do
(7)       if index + 1 > num then
(8)       break;
(9)       end if
(10)      solution_i^p = solution_i^q;
(11) end for
(12) int num = round(ran(0, 1) × solution.length);
(13) int index = 0;
(14)   for int i = 0; i < solution^new.length; i++ do
(15)   if index + 1 > num then
(16)      break;
(17)   end if
(18)   if index + 1 > solution.length then
(19)      index = 0;
(20)      end if
(21)   solution_i^new = solution_i^p;
(22) end for
```

Algorithm 2: Exploitation.

4.3. Pseudocode of Co-Evolutionary Algorithm. Given the maximum generation *maxGen*, Algorithm 3 shows CoEA's pseudocode.

CoEA first initializes all the elements of two PVs as 0.5 and then uses them to initialize two elite solutions $\text{solution}^{\text{elite}_{\text{better}}}$ and $\text{solution}^{\text{elite}_{\text{worse}}}$. In each generation, CoEA, respectively, updates PV_{better} and PV_{worse} through exploration and exploitation strategies. At the end of each generation, if $\text{solution}^{\text{elite}_{\text{worse}}}$'s fitness value is higher than that of $\text{solution}^{\text{elite}_{\text{better}}}$, we will switch them and two PVs. If two PVs' Hamming distance [21] is smaller than 0.5, we will re-initialize it by setting all the elements as 0.5, which ensures the population's diversity. Finally, when reaching the maximum

```
(1)  ********* Initialization **********
(2)  Initialize PV_better and PV_worse by setting all their elements as 0.5;
(3)  solution^elite_better = PV^better. generateSolution();
(4)  generateSolution();
(5)  int index = 0;
(6)     ********** Evolution **********
(7)     int gen = 0;
(8)     while gen < max Gen do
(9)        ********* Update PV^better **********
(10)    solution^new = exploration;
(11)    solution^winner = compete(solution^new, solution^elite_better );
(12)       if solution^winner == solution^new then
(13)       solution^elite_better = solution^new;
(14)    end if
(15)    for int i = 0; i < PV^better.length; i++ do
(16)     if PV_i^better == 1 then
(17)    PV_i^better == PV_i^better + 1/PV_i^better.length;
(18)    else
(19)       PV_i^better == PV_i^better − 1/PV_i^better.length;
(20)    end if
(21)    end for
(22)       ********* Update PV^worse **********
(23)       solution^new = exploitation();
(24)     solution^winner = compete(solution^new, solution^elite_worse );
(25) if solution^winner == solution^new then
(26)       solution^elite_worse = solution^new;
(27)    end if
(28)       for int i = 0; i < PV^worse.length; i++ do
(29)    if PV_i^better == 1 then
(30)    PV_i^better == PV_i^better + 1/PV_i^better.length;
(31)       else
(32)    PV_i^better == PV_i^better − 1/PV_i^better.length;
(33)    end if
(34) end for
(35)       ********* Competition *********
(36)       if solution^elite_worse is better than solution^elite_better then
(37)    switch solution^elite_worse and solution^elite_better ;
(38)    switch PV^worse and PV^better;
(39)    end if
(40)    if HammingDist(PV^better, PV^worse) < 0.5 then
(41)    Initialize PV^worse by setting all the elements as 0.5;
(42)       end if
(43)       gen++;
(44)    end while
(45) return solution^elite_better ;
```

Algorithm 3: Co-evolutionary algorithm.

iteration number $maxT = 3000$, the algorithm terminates and returns $\text{solution}^{\text{elite}_{\text{better}}}$. When the quality of alignment is 1.00, the algorithm can be terminated in advance. However, this situation is barely met in the experiment because it is difficult for the similarity measure to distinguish all the heterogeneous entity pairs. Therefore, this work executes the algorithm until it reaches the maximum generation.

5. Experiment

5.1. Experimental Configuration. In the experiment, Ontology Alignment Evaluation Initiative (OAEI)'s Multifarm track [2], which includes 45 ontology pairs in different languages, is used to test CoEA's performance, and Table 1 gives a brief descriptions on the testing cases.

Table 1: Descriptions on the ontologies in the testing cases.

Testing case	Description
ar-cn	Arabic (ar) vs. Chinese (cn)
ar-cz	Arabic (ar) vs. Chinese (cn)
ar-de	Arabic (ar) vs. German (de)
ar-en	Arabic (ar) vs. English
ar-es	Arabic (ar) vs. Spanish (es)
ar-fr	Arabic (ar) vs. French (fr)
ar-nl	Arabic (ar) vs. Dutch (nl)
ar-pt	Arabic (ar) vs. Dutch (nl)
ar-ru	Arabic (ar) vs. Russian (ru)
cn-nl	Chinese (cn) vs. Dutch (nl)
cn-de	Chinese (cn) vs. German (de)
cn-ru	Chinese (cn) vs. Russian (ru)
cn-fr	Chinese (cn) vs. French (fr)
cn-cz	Chinese (cn) vs. Czech (cz)
cn-pt	Chinese (cn) vs. Portuguese (pt)
cn-es	Chinese (cn) vs. Spanish (es)
cn-en	Chinese (cn) vs. English (en)
cz-de	Czech (cz) vs. German (de)
cz-en	Czech (cz) vs. English (en)
cz-es	Czech (cz) vs. Spanish (sp)
cz-fr	Czech (cz) vs. French (fr)
cz-nl	Czech (cz) vs. Dutch (nl)
cz-pt	Czech (cz) vs. Portuguese (pt)
cz-ru	Czech (cz) vs. Russian (ru)
de-ru	German (de) vs. Russian (ru)
de-es	German (de) vs. Spanish (es)
de-pt	German (de) vs. Portuguese (pt)
de-nl	German (de) vs. Dutch (nl)
de-fr	German (de) vs. French (fr)
de-en	German (de) vs. English (en)
en-es	English (en) vs. Spanish (es)
en-fr	English (en) vs. French (fr)
en-nl	English (en) vs. Dutch (nl)
en-pt	English (en) vs. Portuguese (pt)
en-ru	English (en) vs. Russian (ru)
es-ru	Spanish (es) vs. Russian (ru)
es-pt	Spanish (es) vs. Portuguese (pt)
es-nl	Spanish (es) vs. Dutch (nl)
es-fr	Spanish (es) vs. French (fr)
fr-nl	French (fr) vs. Dutch (nl)
fr-pt	French (fr) vs. Portuguese (pt)
fr-ru	French (fr) vs. Russian (ru)
nl-pt	Dutch (nl) vs. Portuguese (pt)
nl-ru	Dutch (nl) vs. Russian (ru)
pt-ru	Portuguese (pt) vs. Russian (ru)

5.2. Experimental Results. In the experiment, we compare CoEA with EA- and DE-based cross-lingual ontology matching techniques, whose results demonstrated in the tables are the mean values of 30 independent runs. In particular, Table 2 compares CoEA with CEA and CDE in terms of f-measure, and Table 3 carries out T-test [22] among three cross-lingual ontology matching techniques.

Table 2: Comparisons on the alignment's quality in terms of f-measure. The symbols f and std, respectively, stand for mean f-measure and standard deviation.

Testing case	CEA f (std)	HCEA f (std)	CoEA f (std)
ar-cn	0.28 (0.02)	0.28 (0.02)	0.35 (0.01)
ar-cz	0.37 (0.02)	0.37 (0.02)	0.41 (0.01)
ar-de	0.35 (0.01)	0.35 (0.01)	0.43 (0.02)
ar-en	0.36 (0.02)	0.36 (0.01)	0.45 (0.01)
ar-es	0.40 (0.02)	0.40 (0.03)	0.48 (0.01)
ar-fr	0.36 (0.02)	0.36 (0.01)	0.42 (0.02)
ar-nl	0.35 (0.02)	0.35 (0.02)	0.44 (0.01)
ar-pt	0.38 (0.03)	0.38 (0.01)	0.44 (0.01)
ar-ru	0.33 (0.03)	0.33 (0.01)	0.35 (0.02)
cn-cz	0.31 (0.02)	0.31 (0.02)	0.35 (0.01)
cn-de	0.36 (0.02)	0.33 (0.02)	0.41 (0.01)
cn-en	0.30 (0.02)	0.30 (0.01)	0.35 (0.02)
cn-es	0.37 (0.01)	0.37 (0.01)	0.42 (0.01)
cn-fr	0.35 (0.03)	0.35 (0.01)	0.42 (0.01)
cn-nl	0.41 (0.01)	0.41 (0.03)	0.49 (0.02)
cn-pt	0.36 (0.01)	0.36 (0.01)	0.46 (0.01)
cn-ru	0.27 (0.02)	0.27 (0.03)	0.39 (0.01)
cz-de	0.42 (0.02)	0.42 (0.01)	0.45 (0.02)
cz-en	0.44 (0.01)	0.44 (0.03)	0.51 (0.01)
cz-es	0.42 (0.03)	0.42 (0.01)	0.48 (0.01)
cz-fr	0.40 (0.01)	0.45 (0.01)	0.47 (0.01)
cz-nl	0.45 (0.03)	0.42 (0.01)	0.49 (0.02)
cz-pt	0.52 (0.02)	0.44 (0.02)	0.51 (0.01)
cz-ru	0.44 (0.01)	0.42 (0.01)	0.48 (0.01)
de-en	0.42 (0.01)	0.46 (0.01)	0.53 (0.01)
de-es	0.46 (0.02)	0.46 (0.01)	0.55 (0.01)
de-fr	0.46 (0.01)	0.43 (0.03)	0.50 (0.02)
de-nl	0.43 (0.01)	0.41 (0.01)	0.46 (0.01)
de-pt	0.45 (0.03)	0.42 (0.01)	0.53 (0.01)
de-ru	0.42 (0.01)	0.44 (0.02)	0.53 (0.02)
en-es	0.44 (0.02)	0.42 (0.01)	0.49 (0.02)
en-fr	0.42 (0.01)	0.42 (0.01)	0.53 (0.01)
en-nl	0.42 (0.01)	0.45 (0.01)	0.50 (0.01)
en-pt	0.45 (0.01)	0.41 (0.01)	0.51 (0.01)
en-ru	0.45 (0.03)	0.52 (0.02)	0.55(0.01)
es-fr	0.45 (0.02)	0.48 (0.01)	0.55 (0.02)
es-nl	0.48 (0.02)	0.52 (0.02)	0.55 (0.01)
es-pt	0.52 (0.01)	0.52 (0.01)	0.58 (0.01)
es-ru	0.50 (0.03)	0.50 (0.01)	0.63 (0.01)
fr-nl	0.46 (0.01)	0.46 (0.03)	0.52 (0.02)
fr-pt	0.49 (0.01)	0.49 (0.01)	0.58 (0.01)
fr-ru	0.45 (0.01)	0.45 (0.01)	0.55 (0.03)
nl-pt	0.52 (0.02)	0.52 (0.02)	0.58 (0.01)
nl-ru	0.48 (0.01)	0.48 (0.01)	0.58 (0.02)
pt-ru	0.47 (0.03)	0.47 (0.01)	0.51 (0.01)
average	0.41 (0.02)	0.41 (0.01)	0.48 (0.01)

As can be seen from Table 2, CoEA's results are much better than CEA- and HCEA-based ontology matching techniques. CoEA makes use of two different evolving strategies, which is of help to ensures its robustness. From Table 3, we can see that CoEA outperforms other two cross-lingual ontology matching techniques on 5% significant level.

Table 3: T-Test on the alignment's quality.

Testing case	CEA t-value (p value)	HCEA t-value (p value)
ar-cn	93.91 (0.003389)	93.91 (0.003389)
ar-cz	53.66 (0.005931)	53.66 (0.005931)
ar-de	107.33 (0.002966)	107.33 (0.002966)
ar-en	120.74 (0.002636)	190.91 (0.001667)
ar-es	107.33 (0.002966)	75.89 (0.004194)
ar-fr	63.63 (0.005001)	80.49 (0.003954)
ar-nl	120.74 (0.002636)	120.74 (0.002636)
ar-pt	56.92 (0.005592)	127.27 (0.002501)
ar-ru	16.64 (0.019105)	26.83 (0.011858)
cn-cz	53.66 (0.005931)	53.66 (0.005931)
cn-de	67.08 (0.004745)	107.33 (0.002966)
cn-en	53.03 (0.006001)	67.08 (0.004745)
cn-es	106.06 (0.003001)	106.06 (0.003001)
cn-fr	66.40 (0.004793)	148.49 (0.002144)
cn-nl	107.33 (0.002966)	66.56 (0.004782)
cn-pt	212.13 (0.001501)	212.13 (0.001501)
cn-ru	160.99 (0.001977)	113.84 (0.002796)
cz-de	31.81 (0.01)	40.24 (0.007907)
cz-en	148.49(0.002144)	66.40 (0.004793)
cz-es	56.92 (0.005592)	127.27 (0.002501)
cz-fr	148.49 (0.002144)	42.42 (0.007501)
cz-nl	33.28 (0.009561)	93.91 (0.003389)
cz-pt	80.49(0.003954)	93.91 (0.003389)
cz-ru	84.85 (0.003751)	127.27 (0.002501)
de-en	233.34 (0.001364)	148.49 (0.002144)
de-es	120.74 (0.002636)	190.91 (0.001667)
de-fr	53.66(0.005931)	58.24 (0.005465)
de-nl	63.63 (0.005001)	106.06 (0.003001)
de-pt	75.89 (0.004194)	233.34 (0.001364)
de-ru	147.58 (0.002157)	95.45 (0.003334)
en-es	53.03 (0.006001)	93.91 (0.003389)
en-fr	233.34 (0.001364)	233.34 (0.001364)
en-nl	169.70 (0.001876)	106.06 (0.003001)
en-pt	127.27 (0.002501)	212.13 (0.001501)
en-ru	94.86 (0.003355)	40.24 (0.007907)
es-fr	31.81 (0.01)	93.91 (0.003389)
es-nl	93.91 (0.003389)	40.24 (0.007907)
es-pt	127.27 (0.002501)	127.27 (0.002501)
es-ru	123.32 (0.002581)	275.77 (0.001154)
fr-nl	80.49 (0.003954)	49.92 (0.006375)
fr-pt	190.91 (0.001667)	190.91 (0.001667)
fr-ru	94.86 (0.003355)	94.86 (0.003355)
nl-pt	80.49 (0.003954)	80.49 (0.003954)
nl-ru	134.16 (0.002373)	134.16 (0.002373)
pt-ru	37.94 (0.008386)	84.85 (0.003751)

6. Conclusion

To aggregate the ontologies in different languages, this work proposes a CoEA-based cross-lingual ontology matching technique. We first propose a parallel framework of aggregating different SMs and then construct a continuous optimization model to define the problem of cross-lingual ontology integration. To solve this problem, for better trading off the algorithm's exploitation and exploration, we propose a CoEA with two competitive subpopulations. The experiment utilizes OAEI's multifarm track for testing, and the experimental results show that CoEA is able to effectively match cross-lingual ontologies.

In the future, we are interested in further improving the solution's quality by introducing the specific domain knowledge base to distinguish the heterogeneous entity pairs. Moreover, when the scale of ontology becomes large, the searching space of algorithm will definitely grows, which is a challenge for CoEA's efficiency. A feasible approach to face this challenge would be the semantic pruning technique or the ontology partitioning techniques [23], which can be used to improve the efficiency of CoEA when solving large-scale cross-lingual ontologies. [24].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Education Department of Fujian Province (no. JAS160192).

References

[1] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2013.

[2] I. Osman, S. Ben Yahia, and G. Diallo, "Ontology integration: approaches and challenging issues," *Information Fusion*, vol. 71, pp. 38–63, 2021.

[3] M. Abu Helou, M. Palmonari, and M. Jarrar, "Effectiveness of automatic translations for cross-lingual ontology mapping," *Journal of Artificial Intelligence Research*, vol. 55, pp. 165–208, 2016.

[4] X. Xue, J. Chen, and X. Yao, "Efficient user involvement in semiautomatic ontology matching," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 2, pp. 214–224, 2021.

[5] X. Xue, J. Lu, and J. Chen, "Using NSGA-III for optimising biomedical ontology alignment," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 3, pp. 135–141, 2019.

[6] G. Acampora, V. Loia, and A. Vitiello, "Enhancing ontology alignment through a memetic aggregation of similarity measures," *Information Sciences*, vol. 250, pp. 1–20, 2013.

[7] X. Xue and J. Chen, "Optimizing ontology alignment through hybrid population-based incremental learning algorithm," *Memetic Computing*, vol. 11, no. 2, pp. 209–217, 2019.

[8] C. J. Van Rijsbergen, "Foundation OF evaluation," *Journal of Documentation*, vol. 30, no. 4, pp. 365–373, 1974.

[9] S. Mani and S. Annadurai, "Explicit link discovery scheme optimized with ontology mapping using improved machine learning approach," *Studies in Informatics and Control*, vol. 30, no. 1, pp. 67–75, 2021.

[10] X. Xue and Y. Wang, "Optimizing ontology alignments through a memetic algorithm using both MatchFmeasure and unanimous improvement ratio," *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.

[11] X. Xue and Q. Huang, "Generative adversarial learning for optimizing ontology alignment," *Expert Systems*, vol. 2022, pp. 1–12, Article ID 12936, 2022.

[12] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.
[13] G. A. Miller, "WordNet," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[14] E. Geller, M. Gajek, A. Reibach, and Z. Łapa, "Applicability of wordnet architecture in lexical borrowing studies," *International Journal of Lexicography*, vol. 34, no. 1, pp. 92–111, 2021.
[15] X. Xue and J.-S. Pan, "A segment-based approach for large-scale ontology matching," *Knowledge and Information Systems*, vol. 52, no. 2, pp. 467–484, 2017.
[16] M. AlMousa, R. Benlamri, and R. Khoury, "Exploiting non-taxonomic relations for measuring semantic similarity and relatedness in WordNet," *Knowledge-Based Systems*, vol. 212, pp. 1–27, 2021.
[17] J. Chen, X. Xue, Y. Huang, and X. Zhang, "Interactive cross-lingual ontology matching," *IEEE Access*, vol. 7, pp. 79095–79102, 2019.
[18] Y. Xue, H. Zhu, J. Liang J, and A. stowik, "Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification," *Knowledge-Based Systems*, vol. 227, pp. 1–17, 2021.
[19] S.-Z. Zhao and P. N. Suganthan, "Empirical investigations into the exponential crossover of differential evolutions," *Swarm and Evolutionary Computation*, vol. 9, pp. 27–36, 2013.
[20] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.
[21] A. Bookstein, V. A. Kulyukin, and T. Raita, "Generalized hamming distance," *Information Retrieval*, vol. 5, no. 4, pp. 353–375, 2002.
[22] T. K. Kim, "T test as a parametric statistic," *Korean journal of anesthesiology*, vol. 68, no. 6, pp. 540–546, 2015.
[23] X. Xue and J. Zhang, "Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm," *Applied Soft Computing*, vol. 106, pp. 1–11, 2021.
[24] P. Wang, Y. Zhou, and B. Xu, "Matching Large Ontologies Based on Reduction Anchors," in *Proceedings of the Twenty-Second International Joint Conference On Artificial Intelligence*, pp. 1–6, Barcelona Catalonia Spain, July2011.