

A Joint Embedding Method of Relations and Attributes for Entity Alignment

Haihong E, Rui Cheng, Meina Song, Peican Zhu, and Zhen Wang

Abstract—Entity alignment is to link the entities that point to same objects in the real world among different knowledge graphs (KGs). Existing knowledge-embedding-based entity alignment methods mostly regard KG as relation triples, while ignoring attributes and attribute values in KG. However, attribute information provides a valid information supplement for relation triple, alleviates relation triple's relation universality problem and information incompleteness problem, and improves accuracy of entity alignment task. In this paper, we make the first attempt towards combining relation and attribute triples for entity alignment. We divide a KG into relation triples and attribute triples, use parameter sharing (PS) joint method and translation-based knowledge embedding methods to embed them jointly. In addition, we design two strategies: direct accumulation and weight assignment strategy, to explore the effect of relation and attribute triple's embedding on experiment performance. The experimental results show that our method has significantly improved Hits@1, Hits@10 and Mean Rank metrics compared to baseline, and is the state of the arts on entity alignment task. The source code for this paper is available from <https://github.com/ChengRui536/RAKRL>.

Index Terms—Data fusion, data resolution, entity alignment, knowledge graph.

I. INTRODUCTION

Knowledge graph (KG) is a knowledge organization form being adopted to describe various concepts, entities and corresponding relations/attributes in the real world. It is usually expressed as triples, which is further divided into relation triples and attribute ones. The existing largescale KGs, mainly FreeBase [1], DBpedia [2] and Wikidata [3], have been widely used in various applications, e.g., search engines, intelligent assistants, translation systems, question and answer systems and so on.

However, the existing single KG is usually low information coverage, incomplete knowledge description and low knowledge quality, different KGs usually have strong heterogeneity and many knowledge repetitions, which are not conducive to data sharing and integration. So how to fuse different KGs forming a KG with wide knowledge coverage and high knowledge correctness, has become an urgent

This work is supported by the the National Key Research and Development Program of China (2018YFB1403501); Engineering Research Center of Information Networks, Ministry of Education.

Haihong E, Rui Cheng, and Meina Song are with the School of Computer Science, Beijing University of Posts and Telecommunications, China (e-mail: ehaihong@bupt.edu.cn, chengrui536@bupt.edu.cn, mnsong@gmail.com).

Peican Zhu is with the School of Computer Science and Engineering, Northwestern Polytechnical University, China (e-mail: ericcan@nwpu.edu.cn).

Zhen Wang is with the School of Mechanical Engineering, Northwestern Polytechnical University, China (e-mail: w-zhen@nwpu.edu.cn).

problem for the applications based on KGs. Entity alignment is the most critical technology in knowledge fusion.

Some methods have been used for entity alignment task. The most traditional methods are supervised methods, which learn model from annotated data [4], [5]. Nevertheless, these models are typically inextensible and inflexible, and heavily rely on annotated data, thus they are usually time consuming and labor intensive. To avoid these disadvantages, traditional unsupervised methods based on probabilistic methods [6], similarity methods [7] and hierarchical graph models are proposed. However, traditional unsupervised methods usually based on some assumptions or predefined similarity metrics. To compensate for the flaws of supervised methods and traditional unsupervised methods, unsupervised knowledge embedding methods [8]-[13] were proposed. However, these methods all regard KG as relation triples, ignoring attribute ones in it. But attribute information is very useful. For example, for triple (China, area, 9634057), because '9634057' is not a entity, it's not a relation triple, so existing unsupervised knowledge embedding methods will abandon this data, but it's very useful for entity 'china' alignment.

To the best of our knowledge, most of the existed entity alignment methods via knowledge embedding are incapable of distinguishing relation triples from attribute ones.

To solve the problem, we make the first attempt towards combining relation and attribute triples for entity alignment, divide a KG into relation triples and attribute triples, use parameter sharing joint method and translation-based knowledge embedding methods to embed them jointly. Hence, our method is able to fully utilize the internal rich data in KGs without data discarding and flexible to be used in various practical application scenarios.

The main challenges of our method are: how to embed relation and attribute triples separately; how to combine embedding results of relation and attribute triples to achieve better performance. In this paper, according to the characteristics of relation and attribute triples, we use the most main stream and most widely used translation-based knowledge embedding methods (TransE and PTransE) to embed them; and to achieve better performance, we design two strategies for combining embedding results of relation and attribute triples, it's direct accumulate and weight assignment strategy.

The main contributions of this paper are:

- We make the first attempt towards combining relation and attribute triples for entity alignment task, divide KG into relation triples and attribute triples, embed them simultaneously using translation-based knowledge embedding methods, this method fully utilizes the rich internal data of KG, has practicality and applicability.

- Two strategies, i.e., direct accumulation strategy and weight assignment strategy, are designed to explore the effect of relation and attribute triple’s embedding results on entity alignment experiments.
- We evaluated our model in entity alignment tasks, they show that our method has significantly improved Hits@1, Hits@10 and Mean Rank metrics compared to the baseline, and obtained the state of the arts results.

The rest of this paper is organized as follows: Section II briefly introduces some related works; our method is described explicitly in Section III, while the obtained results are illustrated in Section IV, and finally Section V concludes the paper and gives several open lines of future research.

II. RELATED WORK

A. Entity Alignment

Depending on whether use annotated data, methods of entity alignment can be divided into supervised methods and unsupervised methods [14].

Supervised Methods. Supervised methods learn models from annotated data, which mainly include attribute-comparison-based methods, cluster-based methods and active learning methods. [4] proposed a method to automatically generates nonlinear attribute weights to align instances. Nevertheless, Supervised methods heavily rely on annotated datas, and are not extensible and flexible.

Traditional Unsupervised Methods. Traditional unsupervised methods mainly include probabilistic methods, similarity-based methods and hierarchical graph models. [6] proposed a PARIS system based on probabilistic methods, which can align knowledge bases without adjusting parameters and training data, but cannot deal with structural heterogeneity. [7] proposed an unsupervised framework CoLink for user identity link problem. The framework models both attribute based alignment model and relation based alignment model as binary classifiers, and training them in an iterative co-training manner. However, traditional unsupervised methods often require effective metrics based on some assumptions or predefined similarities, with some limitations.

Unsupervised Knowledge Embedding Methods.

Knowledge embedding method encodes information of KG into a continuous lowdimensional semantic space, in which have same meaning or related entities are often close to each other. This method is suitable for largescale KG alignment task. [9] solved cross language or heterogeneous entity alignment problem by jointly embedding global structure information of heterogeneous KGs. However, this method requires rich relational and structural information and does not have applicability. [10] constructed transformations between different vector spaces using TransE for multi-language knowledge alignment. [11] proposed cross KG to simultaneously learn embeddings of two different KGs. [12] proposed a joint embedding method to iteratively align entities, but this method ignores attribute triples. [14] proposed an iterative entity alignment method SEEA based on self-learning and knowledge embedding. This method clearly distinguishes attributes and relations in KG, but only attribute triples are embedded in it’s experiment.

B. Knowledge Embedding

[15] divided knowledge embedding methods into translation distance model and semantic matching model. The former uses a distance-based scoring function, and the latter uses a similarity-based scoring function. This paper mainly investigates the translation distance model including TransE and PTransE. TransE [16] embeds entities and relations into lowdimensional vector spaces, treats r in each triple (h, r, t) as a translation from h to t , makes $h + r \approx t$ as much as possible by constantly adjusting h, r, t . However, this method only considers direct relation of triples in knowledge embedding, ignores rich inference relations that exist between entities. Therefore, [17] proposed a path-based embedding model PTransE, add multistep path embedding between entities to TransE. This paper is based on unsupervised knowledge embedding methods, we use TransE and PTransE for entity alignment experiments.

III. OUR METHOD

We explain the symbols we used in this paper, as shown in Table I.

TABLE I: SYMBOL DESCRIPTION TABLE

Symbol	Description	Symbol	Description
h	head entity	r	relation
t	tail entity	e	entity
a	attribute	v	attribute value
$rt = (h, r, t)$	relation triple	$at = (e, a, v)$	attribute triple
E	set of all entities in a single KG	R	set of all relations in a single KG
A	set of all attributes in a single KG	V	set of all attribute values in a single KG
$RT = \{rt_i\}$	set of all relation triples in a single KG	$AT = \{at_i\}$	set of all attribute triples in a single KG
$KG = (E, R, A, V, RT, AT)$	a complete KG	e'	synonymous entity of entity e
$KG_i = (E_i, R_i, A_i, V_i, RT_i, AT_i)$	different KGs	$\mathbb{L} = (e, e')$	known aligned seeds set

Without loss of generality, we experimented on two KGs which have different structures, they are expressed as $KG_1 = (E_1, R_1, A_1, V_1, RT_1, AT_1)$ and $KG_2 = (E_2, R_2, A_2, V_2, RT_2, AT_2)$. We input \mathbb{L}, RT_1 and RT_2, AT_1 and AT_2 to perform entity alignment task.

A. Overall Structure

Our method has three parts: relation triples joint

embedding part, attribute triples joint embedding part, and co-iterative entity alignment part. Showed as in Fig. 1.

In order to explore the influence of relation triples and attribute triples on entity alignment performance, we design two objective function calculation strategies, namely direct accumulation and weight assignment strategy. If using the direct accumulation strategy, the objective function is defined as:

$$L = R + A + I \quad (1)$$

If using the weight assignment strategy, the objective function is:

$$L = \alpha R + (1 - \alpha)A + I \quad (2)$$

R , A and I represent respectively the score of relation triples joint embedding part, attribute triples joint embedding part and co-iterative entity alignment part, $\alpha \in (0,1)$ represent the weight of relation triples joint embedding part.

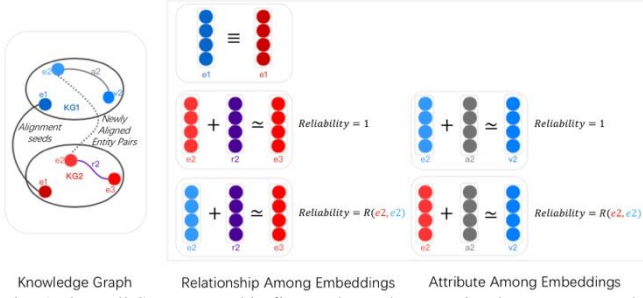


Fig. 1. Overall Structure. This figure shows how we implement our model using TransE. The blue and red points represent entities and attribute values in KG_1 and KG_2 respectively, the gray line represents relations in KG_1 and KG_2 , the purple line represents attributes in KG_1 and KG_2 , the solid line represents the known aligned seeds and the dashed line represents new aligned entity pairs during co-iterative alignment. The same color represents same knowledge and corresponding embeddings.

B. Relation Triples Joint Embedding

In this part, we use parameter sharing (PS) method, TransE and PTransE for relation triples joint embedding. This part can also be implemented by other joint and knowledge embedding methods. We define the score function of this part as:

$$R = P_R + K_R \quad (3)$$

P_R and K_R represent the scores of PS joint model and relation triples embedding method respectively.

Parameter Sharing Joint Model

Since aligned entities have same meaning in different KGs, they can intuitively share same embedding. Formally, foreach aligned entity pair (e, e') , we define $e \equiv e'$. The parameter sharing (PS) model [18] calibrates entities in KG_1 and KG_2 into same semantic space simply and effectively. Therefore the score function of this model is:

$$P_R = 0 \quad (4)$$

Relation Triples Embedding

Based on the PS joint method part, in this part, we use TransE and PTransE to embed relation triples into a semantic vector space.

TransE. TransE [16] treats r in each triple (h, r, t) as a translation from h to t , adjusting h, r, t to make $h + r \approx t$ as much as possible. So the energy function is defined as:

$$E(h, r, t) = \|h + r - t\| \quad (5)$$

We use a margin-based score function as training target, which is defined as:

$$K_{RT} = \sum_{RT \in \{RT_1, RT_2\}} \sum_{(h, r, t) \in RT} L(h, r, t) \quad (6)$$

where $L(h, r, t)$ is a margin-based loss function, defined as follows:

$$L(h, r, t) = \sum_{(h', r', t') \in RT^-} [\gamma + E(h, r, t) - E(h', r', t')]_+ \quad (7)$$

$[x]_+ = \max\{0, x\}$ represents the max number between 0 and x , γ is a margin constant, and RT^- represents a negative sample set of RT , defined as follows:

$$RT^- = \{(h', r, t) | h' \in E\} \cup \{(h, r, t') | t' \in E\} \cup \{(h, r', t) | r' \in R\}, (h, r, t) \in RT \quad (8)$$

Here, RT^- is obtained by randomly replacing one of three components with others in the relation triple (h, r, t) [19].

PTransE. TransE ignores important multistep path information in KG and encounters difficulties in modeling complex relations. For example, (e_1, r_1, e_2) and (e_2, r_2, e_3) can reveal a new fact $(e_1, r_1 \circ r_2, e_3)$, where \circ is a specific function that combines the relations r_1 and r_2 .

Therefore, [17] proposed PTransE, which adds multistep relation path information to TransE. According to PTransE, we define the relation path as: $p = r_1 \circ r_2$ and $P(h, t) = \{p | \forall e \in E, r_1, r_2 \in R, (h, r_1, e), (e, r_2, t) \in RT, p = r_1 \circ r_2\}$. In PTransE, if a relation path and a relation play the same role, that is, $p \in P(h, t)$ and $(h, r, t) \in RT$, we define the relation path embedding as $p \approx r$, it's energy function is:

$$E(p, r) = \|p - r\| = \|p - (t - h)\| = E(h, p, t) \quad (9)$$

PTransE's score function is defined as adding a relation path regularization based on TransE, expressed as:

$$K_{Rp} = K_{RT} + \sum_{RT \in \{RT_1, RT_2\}} \sum_{(h, r, t) \in RT} \left[\frac{1}{Z} \sum_{p \in P(h, t)} R(p|h, t) L(p, r) \right] \quad (10)$$

$Z = \sum_{p \in P(h, t)} R(p|h, t)$ is a normalization factor; $R(p|h, t)$ represents the reliability of path p given entity pair (h, t) ; $L(p, r)$ is a margin-based loss function for (p, r) pairs, defined as:

$$L(p, r) = \sum_{(h, r', t) \in RT^-} [\gamma + E(p, r) - E(p, r')]_+ \quad (11)$$

C. Attribute Triples Joint Embedding

Same as 4.2 section, since attribute triple $at = (e, a, v)$ is composed of entity, attribute and attribute value, there is no multi-step relation path information, so we use PS method and TransE for attribute triples joint embedding. We define the score function this part as:

$$A = P_A + K_A \quad (12)$$

P_A and K_A represent the scores of PS joint model and attribute triples embedding method respectively.

Parameter Sharing Joint Model

Same as the PS joint model in 4.2. The score function is:

$$P_A = 0 \quad (13)$$

Attribute Triples Embedding

Based on PS joint method, in this part, we use TransE to embed attribute triples of different KGs into a semantic vector space.

TransE. Similarly, we also regard a in each triple (e, a, v) as a translation from e to v , by constantly adjusting e, a, v to make $e + a \approx v$ as much as possible. So its energy function is defined as:

$$E(e, a, v) = \|e + a - v\| \quad (14)$$

Also, we use a margin-based score function as training target, which is defined as:

$$K_A = \sum_{AT \in \{AT_1, AT_2\}} \sum_{(e,a,v) \in AT} L(e, a, v) \quad (15)$$

$L(e, a, v)$ is a margin-based loss function, defined as:

$$L(e, a, v) = \sum_{(e', a', v') \in AT^-} [-\gamma + E(e, a, v) - E(e', a', v')]_+ \quad (16)$$

AT^- represents a negative sample set of AT , defined as:

$$AT^- = \{(e', a, v) | e' \in E\} \cup \{(e, a, v') | v' \in V\} \cup \{(e, a', v) | a' \in A, (e, a, v) \in AT\} \quad (17)$$

Similarly, AT^- is derived by randomly replacing one of three components with others in the attribute triple (e, a, v) randomly.

D. Co-iterative Entity Alignment

Based on relation triples joint embedding part and attribute triples joint embedding part, we can perform entity alignment according to semantic distance between entities in the unified semantic space. The semantic distance is defined as: $E(e_1, e_2) = \|e_1 - e_2\|_{L1/L2}, \forall e_1 \in E_1, e_2 \in E_2$. Thus, for an unaligned entity e_1 in a KG, we can find the nearest entity \hat{e}_2 in another KG, i.e.: $\hat{e}_2 = \arg \min_{e_2} (E(e_1, e_2))$. In addition, we use a distance threshold θ , and consider that if $E(e_1, \hat{e}_2) < \theta$, then \hat{e}_2 is very likely to be the aligned entity of e_1 ; otherwise, \hat{e}_2 cannot be the aligned entity of e_1 .

Because relation and attribute triples in same KG have same entity set E , the newly aligned entities can help update relation triples joint embedding space and attribute triples joint embedding space simultaneously, so more aligned entities can be found. Therefore, we use two iterative strategies for joint embedding and entity alignment from baseline [12], it's hard alignment (HA) and soft alignment (SA).

Hard Alignment. This strategy simply uses PS model to newly aligned entities, i.e., $e_2 = e_1$, adds each newly aligned entity pair (e_1, e_2) to the aligned seeds set \mathbb{L} , then updates relation triples joint embedding space and attribute triples joint embedding space to perform further entity alignment. Since the newly aligned entity pair is directly added to \mathbb{L} , it's score function is:

$$I_H = 0 \quad (18)$$

Soft Alignment. When there is a wrong newly aligned entity pair, HA will cause an error accumulation problem. So the SA strategy assigns a reliability score to each newly aligned entity pair, that is, for each newly aligned entity pair (e_1, e_2) , a mapping $R: (e_1, e_2) \rightarrow [0, 1]$ is defined to calculate its reliability score:

$$R(e_1, e_2) = \sigma(k(\theta - E(e_1, e_2))) \quad (19)$$

$\sigma(\cdot)$ is a sigmoid function, k is a hyperparameter satisfying $k \in R^+$.

The score function of SA strategy using direct accumulation strategy is defined as:

$$I_S = I_{S_R} + I_{S_A} \quad (20)$$

While using weight assignment strategy is defined as:

$$I_S = \alpha I_{S_R} + (1 - \alpha) I_{S_A} \quad (21)$$

where,

$$I_{S_R} = \sum_{(e_1, e_2) \in \mathbb{M}} R(e_1, e_2) (H_{R(e_1, e_2)} + H_{R(e_2, e_1)}) \quad (22)$$

$$H_{R(e_1, e_2)} = \sum_{(e_1, r, t)} U(e_2, r, t) + \sum_{(h, r, e_1)} U(h, r, e_2) \quad (23)$$

$$I_{S_A} = \sum_{(e_1, e_2) \in \mathbb{M}} R(e_1, e_2) (H_{A(e_1, e_2)} + H_{A(e_2, e_1)}) \quad (24)$$

$$H_{A(e_1, e_2)} = \sum_{(e_1, a, v)} U(e_2, a, v) \quad (25)$$

$\mathbb{M} = \{(e_1, e_2)_i\}$ denotes newly aligned entity pairs set, $U(h, r, t)$ and $U(e, a, v)$ respectively represent the loss function of relation triples and attribute triples. For TransE and PTransE, we have different U : In TransE, $U(h, r, t) = L(h, r, t)$, $U(e, a, v) = L(e, a, v)$. In PTransE, $U(h, r, t) = L(h, r, t) + \frac{1}{Z} \sum_{p \in P(h, t)} R(p|h, t) L(p, r)$.

IV. EXPERIMENTS

A. Baseline

We chose [12] as our baseline. Their experiments include ITransE(HA), ITransE(SA), IPTransE(HA) and IPTransE(SA), corresponding the adoption of TransE/PTransE, hard alignment strategy (HA)/softalignment strategy (SA) for experiments respectively. Similarly, in our experiment, we have RATransE/RAPTransE(HA/SA- α : $1-\alpha$), which means TransE/PTransE, HA/SA, weight assignment strategy/direct accumulation strategy are adopted respectively in experiments. Here, α : $1-\alpha$ represents the proportion of relation triples to attribute triples in the weight assignment strategy, while there is no α for direct accumulation strategy.

B. Evaluation Metrics

There are two evaluation metrics for entity alignment task: (1) average order of the correct entity (i.e., Mean Rank); (2) correct answer ranked in the top 10 and the top 1 (i.e., Hits@10 and Hits@1). Usually, higher Hits@10, Hits@1 and lower Mean Rank are preferred which indicates better alignment.

C. Datasets

The relation triple dataset in this paper is from FB15k [16], which contains 14,951 entities, 1,345 relations and 592,213 triples. The attribute triple dataset is from FB15K-237 [20], which is a subset of FB15k, with train set, validation set and test set, respectively containing 272,115, 17,535, and 20,466 triples. We named the relation triple dataset as RDFB and the attribute triple dataset as ADFB.

RDFB. We construct this dataset by randomly dividing the triples in FB15k into two same size subsets RT1 and RT2, and making the overlap ratio $O=0.5$ between RT1 and RT2. The entity set E and relation set R in two KGs are same, and the alignment seed set L is selected from the most common entities.

ADFB. We integrated train set, validation set and test set

inFB15K237 to form a dataset which contains 310,116 triples, 310,116 attributes and 310,116 attribute values. Same as the RDFB dataset, we randomly divide these attribute triples into two same size subsets AT_1 and AT_2 , and make the overlap

ratio $O=0.5$ between AT_1 and AT_2 . The entity set E , attribute set A , and attribute value set V in two KGs are same. The entity set E of RDFB and ADFB is same.

The settings of datasets are shown in Table II.

TABLE II: DATASET SETTING TABLE

Dataset	R	E	RT_1	RT_2	A	V	AT_1	AT_2	\mathbb{L}	O
RDFB	1345	14951	444159	444160	-	-	-	-	5000	0.5
ADFB	-	14951	-	-	310116	310116	232587	232587	-	0.5

D. Experiment Settings

For comparison, we followed the experiment settings in baseline, using stochastic gradient descent (SGD) as the optimizer; knowledge embeddings $E=\{e|e\in E\}$, $R=\{r|r\in R\}$, $A=\{a|a\in A\}$ and $V=\{v|v\in V\}$ are initialized to a normal distribution; the dissimilarity measure in TransE is realized by L1 norm; all models are based on same dimension $n=50$ and epochs=3000, and embeddings of entities, relations, attributes, and attribute values are all the same dimensions. For the hyperparameters, we set $\gamma=1.0$, $k=1.0$, learning rate $\lambda=0.001$, for hard alignment, $\theta=1.0$, and soft alignment $\theta=3.0$. We set every 500 iterations to perform soft alignment starting from the 1000th iteration.

E. Experiment Results

Here, the obtained experiment results are illustrated in

Table III. As unveiled, we can draw the following conclusions: (1) Our method significantly improves entity alignment performance, and achieves the state of the arts results. This shows that using attribute triples in KG will significantly improve entity alignment performance. (2) The results of direct accumulation strategy and weight assignment strategy are not much different. This may be because relation triples and attribute triples are initialized same when knowledge embedding, and optimized method is also the same, so the difference between them is small. (3) The PTransE based approach outperforms the corresponding TransE based approach, indicating that more comprehensive knowledge embedding will result in more accurate alignment. (4) Soft alignment strategy is better than the corresponding hard alignment strategy, this maybe because hard alignment strategy cause error accumulation problem.

TABLE III: ENTITY ALIGNMENT RESULTS

	Hits@1	Hits@10	Mean Rank		Hits@1	Hits@10	Mean Rank
ITransE(SA)	67.1	83.1	80.1	IPTransE(SA)	71.7	86.5	49.0
RATransE(SA)	73.5	86.6	57.8	RAPTransE(SA)	77.0	89.5	40.5
RATransE(SA-0.5:0.5)	73.3	86.7	54.2	RAPTransE(SA-0.5:0.5)	77.5	89.7	36.1
RATransE(SA-0.9:0.1)	73.1	86.5	55.4	RAPTransE(SA-0.9:0.1)	76.7	89.0	38.1
RATransE(SA-0.1:0.9)	73.0	86.5	58.3	RAPTransE(SA-0.1:0.9)	76.0	89.1	38.9
ITransE(HA)	62.6	78.9	100.0	IPTransE(HA)	66.1	83.3	59.1
RATransE(HA)	68.1	84.4	68.0	RAPTransE(HA)	68.4	87.1	46.1
RATransE(HA-0.5:0.5)	68.2	84.3	69.1	RAPTransE(HA-0.5:0.5)	68.9	86.6	49.4
RATransE(HA-0.9:0.1)	68.2	84.0	67.8	RAPTransE(HA-0.9:0.1)	68.1	86.0	48.5
RATransE(HA-0.1:0.9)	68.2	84.1	68.7	RAPTransE(HA-0.1:0.9)	68.8	86.7	52.4

TABLE IV: EXPERIMENT RESULTS AT DIFFERENT ITERATIONS

epochs	500			1000			1500		
	Hits@1	Hits@10	Mean Rank	Hits@1	Hits@10	Mean Rank	Hits@1	Hits@10	Mean Rank
ITransE(HA)	54.9	75.9	101.7	57.6	77.8	91.9	59.2	78.2	95.7
RATransE(HA)	62.8	81.5	72.6	65.9	83.1	70.5	66.7	83.4	67.9
ITransE(SA)	55.0	75.6	96.1	58.1	77.6	90.0	62.2	80.2	85.8
RATransE(SA)	62.9	81.3	73.1	66.2	83.3	67.1	69.2	84.8	66.5
epochs	2000			2500			3000		
	Hits@1	Hits@10	Mean Rank	Hits@1	Hits@10	Mean Rank	Hits@1	Hits@10	Mean Rank
ITransE(HA)	60.8	78.8	93.6	61.0	78.6	97.9	60.8	78.6	100.1
RATransE(HA)	67.6	83.8	71.8	68.2	84.3	70.9	68.1	84.4	68.0
ITransE(SA)	65.1	81.4	81.9	66.6	82.2	82.8	67.1	83.0	79.2
RATransE(SA)	70.8	85.7	63.4	72.2	86.0	58.8	73.5	86.6	57.8

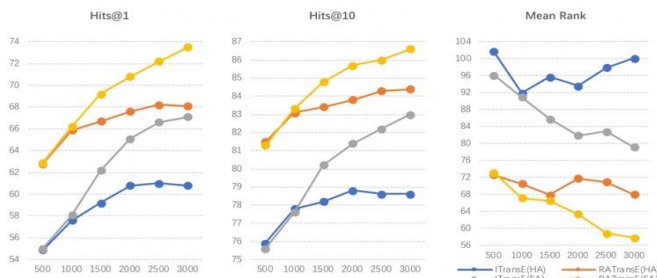


Fig. 2. Experiment results at different iterations.

Furthermore, we show the results of baseline method and our method under different iterations in Table IV and Fig. 2, we find that: (1) Regardless of iterations number, our method is better than baseline method correspondingly. Even HA model of our method is better than SA model of baseline method, the results of our method at 1000th epoch is better than baseline method at 3000th epoch. These prove that the adding of attribute triples is superior to the transformation of iterative alignment strategies and the increase of iteration

number. (2) The performance of all methods improves with the number of iterations increasing. From the 500th iteration to the 3000th iteration, the metrics of soft alignment strategy have a relatively stable upward or downward trend, while hard alignment's metrics are unstable. This may be due to hard alignment's error accumulation problem. To balance experiment's performance and efficiency, we provide 3000th iteration results finally.

V. CONCLUSION AND FUTURE WORK

This paper makes the first attempt towards combing relation and attribute triples for entity alignment task, by dividing KG into relation triples and attribute triples, using parameter sharing joint method and translation-based knowledge embedding methods to embed them jointly. Moreover, to verify the influence of relation and attribute triples' embedding results on experiment performance, we design direct accumulate strategy and weight assignment strategy. The experimental results show that our method obviously improves the performance of entity alignment task, and is the state of the arts. The source code for this paper can be obtained from <https://github.com/ChengRui536/RAKRL>.

In the future, we will explore the following research directions: (1) consideration of the rich external information in KGs (such as descriptive text) for entity alignment; (2) application of our model in real medical knowledge graph entity alignment tasks; (3) analyzing the performance of other effective knowledge embedding models with the adoption of our method.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Haihong E, Rui Cheng and Meina Song concuted the research design; Haihong E and Rui Cheng completed the research implementation; Haihong E and Rui Cheng analyzed the data; Haihong E, Rui Cheng, Peican Zhu and Zhen Wang wrote the paper. all authors had approved the final version.

REFERENCES

- [1] B. Kurt, E. Colin, P. Praveen, S. Tim, and T. Jamie, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247-1250.
- [2] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch *et al.*, "DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, pp. 167-195, 2015.
- [3] V. Denny and K. Markus, "Wikidata: A free collaborative knowledgebase," *Commun. ACM*, pp. 78-85, 2014.
- [4] M. H. Seddiqui, R. P. D. Nath, and M. Aono, "An efficient metric of automatic weight generation for properties in instance matching technique," in *Proc. CoRR*, 2015.
- [5] N. Maximilian, T. Volker, and K. H. Peter, "A three-way model for collective learning on multi-relational data," in *Proc. the 28th International Conference on International Conference on Machine Learning*, 2011, pp. 809-816.
- [6] M. S. Fabian, A. Serge, and S. Pierre, "PARIS: Probabilistic alignment of relations, instances, and schema," in *Proc. VLDB Endow.*, 2011, vol. 5, no. 3, pp. 157-168.
- [7] Z. Zhong, Y. Cao, M. Guo, and Z. Nie, "CoLink: An Unsupervised Framework for User Identity Linkage," in *Proc. Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

- [8] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proc. CoRR*, 2014.
- [9] H. Yanchao, Z. Yuanzhe, H. Shizhu, L. Kang, and Z. Jun, "A joint embedding method for entity alignment of knowledge bases," in *Proc. Knowledge Graph and Semantic Computing: Semantic, Knowledge, and Linked Big Data*, 2016, pp. 3-14.
- [10] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," in *Proc. the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 1511-1517.
- [11] P. Cai, W. Li, Y. Feng, Y. Wang, and Y. Jia, "Learning knowledge representation across knowledge graphs," in *Proc. Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [12] H. Zhu, R. Xie, Z. Liu, and M. Sun, "Iterative entity alignment via joint knowledge embeddings," in *Proc. the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 4258-4264.
- [13] N. Maximilian, R. Lorenzo, and P. Tomaso, "Holographic embeddings of knowledge graphs," in *Proc. the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1955-1961.
- [14] S. Guan, X. Jin, Y. Jia, Y. Wang, H. Shen, and X. Cheng, "Self-learning and embedding based entity alignment," in *Proc. 2017 IEEE International Conference on Big Knowledge (ICBK)*, 2017, pp. 33-40.
- [15] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: a survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724-2743, 2017.
- [16] B. Antoine, U. Nicolas, G. D. Alberto, W. Jason, and Y. Oksana, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2787-2795, 2013.
- [17] Y. Lin, Z. Liu, and M. Sun, "Modeling relation paths for representation learning of knowledge bases," in *Proc. CoRR*, 2015.
- [18] G. Ian, B. Yoshua, and C. Aaron, "Deep learning," The MIT Press, 2016.
- [19] B. Antoine, G. Xavier, W. Jason, and B. Yoshua, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, pp. 233-259, 2014.
- [20] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," *EMNLP*, 2015.

Copyright © 2020 by the authors. 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 ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).



Haihong E was born in 1982 in Liaoning Province, China. She is an associate professor and Ph.D at the School of Computer Science, Beijing University of Posts and Telecommunications. Her research's area is in deep learning, knowledge mapping, natural language processing, big data and artificial intelligence.



Rui Cheng was born in 1995 in Shanxi Province, China. She is at Beijing University of Posts and Telecommunications, School of Computer Science. Her research's area is in knowledge graph and natural language processing.



Meina Song was born in 1974 in Shandong Province, China. She is a professor and Ph.D at the School of Computer Science, Beijing University of Posts and Telecommunications. Her research's area is in medical knowledge graph, artificial intelligence and data mining



Peican Zhu was born in 1985 in Henan province, China. He received the B.S. and M.Sc. degrees from Northwestern Polytechnical University (NWPU), Xi'an, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Alberta, Edmonton, Canada, in 2015. He is currently an associate professor with the School of Computer Science and Technology, NWPU. His current research interests include fault diagnosis and system behavior prediction, approximate computing, gene regulatory networks, complex networks, network disease, machine learning and big data.



Zhen Wang received the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2014. From 2014 to 2016, he was a JSPS senior researcher with the Interdisciplinary Graduate School of Engineering Sciences, Kyushu University, Fukuoka, Japan. Since 2017, he has been a full professor with Northwestern Polytechnical University, Xi'an, China. Thus far, he has published more than 100 scientific papers and obtained more than 10000 citations. His current research interests include network science, complex system, big data, evolutionary game theory, behavior decision, and behavior recognition. He was a recipient of the National 1000 Talent Plan Program of China. He serves as an editor or an academic editor for seven journals.