Entity alignment via joint GCN structure information embedding and TransE attribute information embedding*

Haihong E[a], Rui Cheng[a,*], Jingru Liang[a] and Meina Song[a]

[a]Beijing University of Posts and Telecommunications, 10 Xitucheng Road, Haidian District, Beijing

ARTICLE INFO

Keywords:
Entity Alignment (EA)
Knowledge Graph (KG)
GCN
TransE

ABSTRACT

Entity Alignment (EA) links entities that exist in different knowledge graphs (KGs) but point to the same objective objects in the real world. The mainstream EA methods can be generally divided into relation based learning methods, relation and attribute based methods, and structure and attribute based methods. However, in these methods, the same model or method is used to learn the structure and attribute information at the same time, resulting in that both the structure and attribute information cannot be completely learned. In this paper, we use different embedding methods to learn structure and attribute information respectively so that we can preserve the structure and attribute information to the greatest extent. Considering that KG has complex structural information, and GCN can fully mine structural information, we use GCN to embed structural information to retain the relationship dependencies between entities. Since attribute information is sequence information which does not include structure information, and TransE based on translation rules has a good effect on embedding sequence information, we use TransE to embed attribute information. In addition, we propose three joint strategies to combine the embedding results of structure and attribute information for EA. Experimental results show that our methods have improved the metrics of hits@1, hits@10 and hits@50 by an average of 10.51%, 10.37%, and 7.57% respectively compared with the best baseline methods. The source code for this paper is available at `https://github.com/ChengRui536/TransE-GCN`.

1. Introduction

Knowledge Graph (KG) is a form of knowledge organization used to describe various concepts, entities, relationships and attributes in the real world. It is usually expressed as triples, which can be further divided into relation and attribute triples. The existing large KGs are mainly FreeBase[1], DBpedia[9], and Wikidata[17], which have been widely used in various applications, such as search engines, smart assistants, translation systems, and question answering systems.

However, existing single KG usually has low information coverage, incomplete knowledge description, and low knowledge quality. But different KGs usually have strong heterogeneity and high knowledge repetition, which are not conducive to data sharing and integration. Therefore, how to integrate different knowledge to form a KG with broad knowledge and high knowledge correctness has become an urgent issue for KG-based applications. Entity alignment (EA) is the most important and critical technique in knowledge fusion.

Some methods have been used for EA tasks. The traditional methods learn the relational triples of entities. For example, the TransE[2] model models the relational triples (s, r, o) as $s+r \approx o$. Later on the basis of TransE, TransH[19], TransR[10], TransD[8] and TransA[21] were proposed. The JE[7], IPTransE[25], MTransE[5], BootEA[15], SEA[12], and KAGAN[13] were proposed later which also utilized EA based on relational triples.

However, the traditional methods of learning relational triples only consider 1-hop neighbor information, with low expression ability and insufficient information utilization. The RSN[6] proposed in 2019 attempts to use a cyclic hopping network to learn relationship paths between entities and relations, which relatively improves the shortcomings of traditional methods. However, the methods still cannot capture the complex structure information in KG.

Graph neural networks (GNN) excel in obtaining structural information of graphs. In the past two years, using GNN to represent entities in KG has gradually become a new direction for EA task. Wang et al.[18] proposed a graph convolutional network (GCN) to perform EA (GCN-Align), which proved that GCN can process multi-relational information and is superior to traditional ones in capturing complex graph structures for the first time. Pang et al.[11] proposed INGA, which is an iterative training model based on GCN by combining structure and attribute information. Hybrid Multi-Aspect Alignment Networks (HMAN)[22] uses the bag-of-words model to initialize entity structure, relation and attribute information, and regards them as the inputs of GCN to learn embeddings. However, the above methods cannot fully retain the information of structure and attribute in KG, because they usually utilize the same model to learn them, which results in low quality of EA task.

As far as we know, the existing methods either cannot fully utilize the attribute and structural information in KG, or use the same method to model all the information. Therefore, low quality of information learned from KG and poor EA performance are caused.

In order to solve these problems, in this paper, we learn to embed KG's structure and attribute information at the same

*This document is the results of the research project funded by the National Natural Science Foundation of China (No. 61902034); Engineering Research Center of Information Networks, Ministry of Education.

*Corresponding author

ehaihong@bupt.edu.cn (H. E); chengrui536@bupt.edu.cn (R. Cheng); liangjingru@bupt.edu.cn (J. Liang); mnsong@gmail.com (M. Song)

ORCID(s):

time. Aiming at the characteristics of attribute information and structure information in KG, we will 1) use different methods to learn the attribute and structure information respectively; 2) design multiple combination strategies to integrate the learned results of attribute and structure information; and 3) perform the EA task.

The main challenges of our method are: 1) in terms of learning the different characteristics of structure and attribute information in KG, it is difficult to select the appropriate methods; and 2) in order to obtain a better performance, it is difficult to design the joint strategies, to combine the embedding results of structure and attribute information.

For the first challenge, we use the embedding methods that can preserve structural information and attribute information to the greatest extent. Because the structural information in KG is multi-step and complex, and GCN is a neural network specially designed for graph structure, it can fully mine the structural information of KG. So we use GCN to embed structural information to preserve the relationship between dependent entities. For the attribute information, since it is sequence structure information which does not contain the graph structure information, and TransE has a good effect on embedding sequence information, we use TransE to embed attribute information. In addition, we propose three joint strategies, i.e., connection strategy, weight allocation strategy, and iterative strategy, to tightly combine the embedding results of structural information and attribute information for EA.

The main contributions of this paper are:

- We combine the classic method TransE and the graph calculation method GCN for EA. Because GCN can fully mine the structure information of KG, we use GCN to embed structure information to preserve the relationship dependencies between entities. Since TransE is based on translation rules which has a good effect on the embedding of sequence information, we use it to embed attribute triples.

- Aiming at integrating the embedding results of the structure and attribute information, we propose three joint strategies, namely connection strategy, weight allocation strategy, and iterative embedding strategy, to explore the effect of each strategy on the EA task.

- We evaluated our method in EA task and the results show that our methods have improved the metrics of hits@1, hits@10 and hits@50 by an average of 10.51%, 10.37%, and 7.57% respectively compared with the best baseline methods.

The rest of this paper is organized as follows: Section 2 briefly introduces some related work. Our method is clearly described in Section 3, and the experimental results are explained in Section 4. Finally, in Section 5 we make the summary and discuss the directions for the future research.

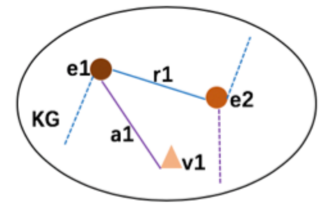


Figure 1: KG Structure. The circle represents the entity, the triangle represents the attribute value, the blue line represents the relationship between the entities, the purple line represents the attribute connection between the entity and the attribute value.

2. Preliminaries

Entity refers to the object that exists in the real world. The relationship is used to describe the connection between two entities. The value of the connection is an entity, which is a manifestation of the entity's external association. Attributes are the inherent characteristics of an entity. The connection between the entity and the attribute value, is an internal manifestation and internal association of the entity, and does not connect with other entities. The attribute value is a text description.

The KG structure discussed in this paper mainly includes structural information (relational triplet) and attribute information (attribute triplet), where the relationship triplet takes the form of <entity, relationship, entity>; the attribute triplet takes the form of < Entity, attribute, attribute value>. As shown in Figure 1.

Entity Alignment is also called Entity Matching and Entity Linking. It refers to each entity in the knowledge graph of heterogeneous data sources, according to the relevant information of each entity, to find out which belongs. The same objective objects in the real world are linked together, and the linked entities are called aligned entities. Entity alignment refers to using multiple information of the entity itself to determine whether two or more entities point to the same object in the objective world.

In the entity alignment task, the fundamental basis for judging whether the entities in different knowledge graphs are aligned is the various information possessed by the entities in the knowledge graph. With the similarity of multiple information, it can be judged whether the entities are aligned.

3. Related work

According to the different uses of KG information, EA methods can be divided into relationship learning based, relationship and attribute joint learning based, and structure information learning based graph neural network (GNN).

3.1. Relationship learning based methods

The EA methods based on relationship learning mainly use relation triples in KG to perform EA, which is formed as

<head entity, relationship, tail entity>. Hao et al.[7] proposed a JE model to perform EA by combining two pre-matched entities in KGs, training relational triples to adjust entity embeddings, and learning entity and relation embeddings into a unified vector space. Zhu et al.[25] proposed IP-TransE. They considered relation paths between entities during training and used iteration and parameter sharing methods for EA. MTransE[5] was proposed by Chen et al. It was a translation-based Multilingual KG embedding model, which encoded entities and relationships in KGs into separate embedding spaces, and considered three techniques, i.e., distance-based axis calibration, translation vector, and linear transformation, to learn cross-language transformation of entities and relationships in different vector spaces. BootEA[15] was proposed by Sun et al. They labeled possible aligned entities as training data during the training phase, and allowed the matched entities to be edited or deleted to further mitigate error propagation. In SEA[12], proposed by Pei et al., they represented entities' multiple relations as degree, and used adversarial training to improve the degree difference problem of KG embedding. Qu et al.[13] proposed KAGAN, which was a weakly supervised EA model based on adversarial learning and mutual information maximization. Guo et al. proposed RSNs[6]. They used a network hopping mechanism to obtain entity long-term dependencies. However, the above methods only used the relationship information in KG for EA, and could not capture the structural and attribute information in KG. In addition, they ignored the effect of attribute and structural information on EA task.

3.2. Relationship and attribute joint learning based methods

Later, researchers found that attributes can be of great help to EA task. On the one hand, the relationships between entities may be sparse, and learning only the relationships may lead to insufficient effective information problem for EA. On the other hand, entities of the same type generally have similar types of relationships, that is, the relationships are universal, but their attributes and attribute values are generally different. So entities are more likely to match. JAPE[14] was proposed by Sun et al. They embedded entities and relations in KG into the same vector space, and then used the attribute information to improve the embedding results. Chen et al.[4] proposed a collaborative training-based method KDCoE, which iteratively trained two embedded models on the relationship and attribute level in multilingual KG to perform EA. Trisedya et al.[16] proposed a new method, which used attribute character sequences to transform the attributes information into natural language semantic level embeddings for EA, and solved the problem of using attribute similarity methods to perform EA but without attributes in entities. Zhang et al.[24] proposed a multi-view based EA framework MultiKE which trained the name, relationship, and attribute views on different models, and combined entity representations of multiple views to complete EA task. The experimental results of the above models show that entity representations obtained by joint learning of relationships and attributes in KG can complete a higher-quality EA task.

3.3. Structure information learning based graph neural network (GNN) methods

The representation of KG based on GNN method is a new research direction and the hot spot in recent years. Some researchers have put forward the following opinions: (1) equivalent entities often have similar relationships; (2) equivalent entities often have equivalent neighbors. This is in line with the idea of graph convolutional network (GCN). GCN was first applied to KG's representation, when Wang et al.[18] proposed an EA method based on GCN. The paper first proved that GCN can perform convolution operations on KGs to generate neighbor-adaptive entity embeddings, so it can be used for EA tasks. Later, Ye et al. proposed VR-GCN[23]. It was a relational vectorized GCN method for multi-relational network embedding, and explicitly learned the relationship representation in KG. This method can help to understand the structure of KG. The RDGCN[20] was proposed by Wu et al.. They introduced the concept of dual graphs. Through the interaction between the dual graph and the original KG, the paper has fully learned the relationship information in KG to obtain a better entity representation. Cao et al.[3] proposed MuGNN, a multi-channel graph attention network. The paper first completed the KG, then generated adjacency matrices based on self-attention and cross-graph attention mechanism. Afterwards, it used them as the input into GCN to encode entities. But the above methods all ignore attribute information in KG. More recently, researchers tried to use GCN to solve this problem. Pang et al.[11] proposed INGA, which is an iterative training model based on GCN combining structure and attributes information. Yang et al.[22] proposed the HMAN method, which combined multiple aspects of entities in KG based on GCN, i.e., topological connections, relationships, attributes, and entity descriptions. The above methods utilized a wealth of information such as relationships, attributes, or descriptions of entities in KG, to improve the effect of EA. However, in these methods, the same method was used to learn the structure and attribute information at the same time, which results in the fact that both the structure and attribute information could not be completely learned. Therefore these methods have a limited improvement in EA task.

4. Our method

We explain the symbols we use in this paper, as shown in Table 1.

Without loss of generality, we perform experiments on two different language KGs, which 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 enter $\mathbb{L}$, RT_1 and RT_2, AT_1 and AT_2 to perform the EA task.

Table 1
Symbol Description Table

Symbol	Description	Symbol	Description
h	head entity	r	relation
t	tail entity	e	entity
a	attribute	v	attribute value
$rt = (h, r, t)$	structural information	$at = (e, a, v)$	attribute information
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 structural information in a single KG	$AT = \{at_i\}$	set of all attribute information 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

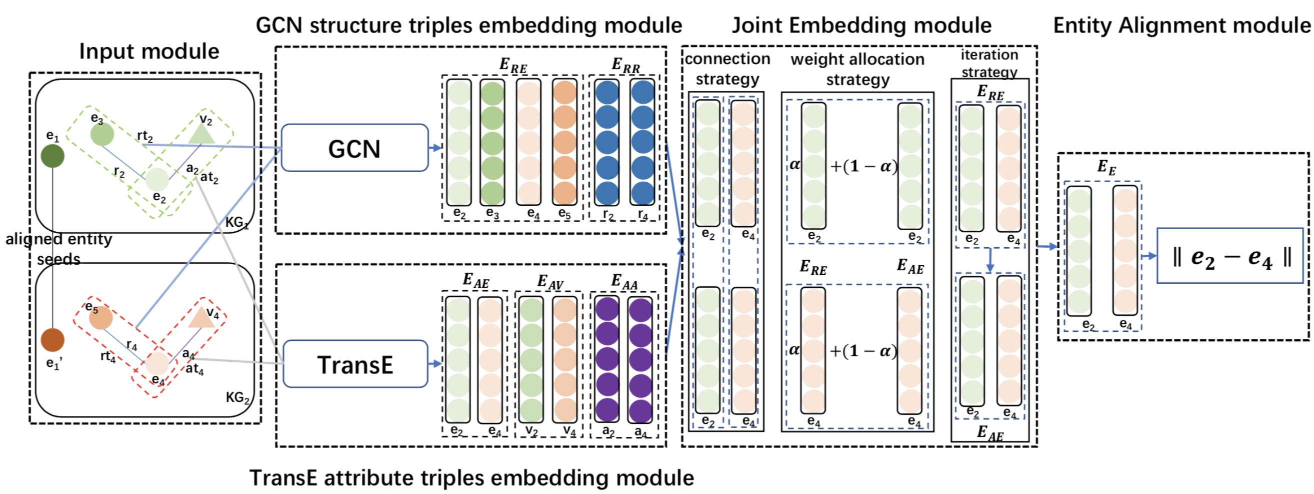


Figure 2: Overall Structure. The overall structure includes 4 parts. (1) The first part of the input module: two or more KGs, and a small number of aligned entity sets between multiple KGs (used as a training seed set) are utilized as the input; (2) Part 2 independently embeds the module, including GCN structure information embedding module and TransE attribute information embedding module. GCN structure information embedding module uses the relationship information data in the input module to finally get the entity embedding result (E_{RE}) and relationship embedding result (E_{RR}) in the relationship data; TransE attribute information embedding. The module uses the attribute information data in the input module to finally get the entity embedding result (E_{AE}), attribute embedding result (E_{AA}) and attribute value embedding result (E_{AV}) in the attribute data; (3) the third part is the embedding joint module, which has been designed three different strategies to calculate the entity embedding result obtained by the independent embedding module, to obtain the final entity embedding result; (4) Part 4 is the entity alignment module, which calculates whether the entities in the KGs are aligned according to the entity embedding result or the entity alignment calculation method.

4.1. Overall structure

Our method includes three parts: GCN structure information embedding part, TransE attribute information embedding part and embedding joint strategy part. The overall architecture is shown in Figure 2.

4.2. GCN structure information embedding

GCN has been proven that it can effectively capture complex structural information from KGs; and equivalent entities tend to be adjacent to other equivalent entities through the same type of relationship in KG. Therefore, we use GCN to effectively capture the complex structural information which can be commonly found in multi-relational KGs. Here, we use GCN to project the structure information, which contains entity, relationship and structure information in KG, into a low-dimensional vector space, where equivalent entities are close to each other. The framework is shown in Figure 3.

The GCN model is a stack of multiple GCN layers to collect information from multi-hop neighbors. The GCN inputs entity feature X^l into the $l+1$ GCN layer, and outputs entity representation $X^{(l+1)}$, which is calculated as follows:

$$X^{(l+1)} = \xi(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} X^{(l)} W^{(l)}) \tag{1}$$

Where ξ is the RELU activation function. $\hat{A} = A + I$, where A represents the relationship matrix of structural information in KG. I is the identity matrix, and $\hat{D}_{jj} = \sum_k \hat{A}_{jk}$ is

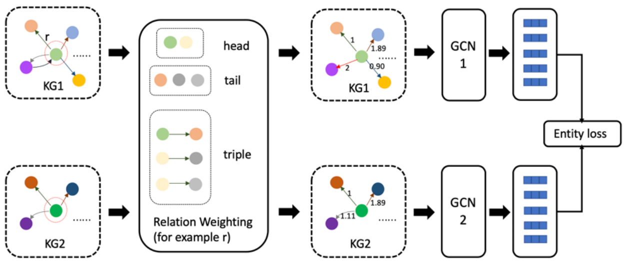


Figure 3: The illustration of using GCN to embed structure information. In KG1 and KG2, the solid circles with different colors represent different entities, and the lines with different colors represent different relationships. We focus on the entities that are surrounded by the red hollow circles. KG1 and KG2 were weighted respectively to obtain weighted graphs KG1 and KG2 first; then they were utilized as the input to GCN1 and GCN2 respectively, and finally the entity and relationships embeddings were learned through convolution operations on the graph.

the node degree matrix of $\hat{A}$. $W^{(l)} \in R^{d(l)*d(l+1)}$ is a trainable weight matrix of specific layer.

Weighting relationships. Since the traditional GCN can only handle undirected uni-relationship networks, it is not suitable for knowledge graphs with multiple relationships. In the traditional GCN, $A_{ij} \in A$ indicates whether nodes i and j are adjacent, which obviously cannot be used to express the rich relationship between entities in the knowledge graph. In the knowledge graph, the relationships between entities are represented by edges. Since the probability that two entities are equivalent depends largely on the relationship between them and the aligned entities, we first need a measurement method to measure the relationship. The higher the proportion of head (tail) entities connected to a relationship, the greater the value of the measured relationship. So we weight the relationship between the entities. We weight the relationship between the entities, measured by A_{ij}, which represents the degree of influence of entity i on entity j. Each relationship is measured by a head entity, a tail entity, and a triple, which are called functional and inverse functional calculations. The calculation is formulated as follows:

$$deg(r) = \frac{h\_num_r}{tri\_num_r} \tag{2}$$

$$rdeg(r) = \frac{t\_num_r}{tri\_num_r} \tag{3}$$

Among them, tri_num_r is the number of relationship triples existing in relationship r, and h_num_r and t_num_r are the number of types of head entities and tail entities in the relationship triples where relationship r exists. We believe that the probability that two entities are equivalent depends largely on the relationship between them and the pair of known aligned entities. Therefore, we define A_{ij} to represent the degree of influence of entity i on entity j, measured by the values of all relationships between entity i and entity j calculated by the formula in the following:

$$A_{ij} = \sum_r rdeg(r) + \sum_{r'} deg(r') \tag{4}$$

where $r \in \{r, (e_i, r, e_j) \in KG\}$,$r' \in \{r, (e_j, r, e_i) \in KG\}$.

Relationship embedding. In this paper, we use GCN1 and GCN2 to embed entities in the cross-language KG1 and KG2 into the same vector space respectively. GCN1 and GCN2 models share two layers' weight matrix W. Each entity in the GCN layer has a relational feature vector, which is initialized randomly at layer 0 and continuously updated during training. The final outputs of GCN1 and GCN2 are the cross-language KGs' entities embedding result, which contains the relationship information and structure information in two KGs. This embedding result will be combined with the entity embedding result which contains attribute information to perform EA tasks.

Loss function. In order to make the distance of equivalent entities in the vector space be as small as possible and non-equivalent entities as large as possible, we designed the following loss function. We used the entity pair S in the seed entity as training data to train the GCN1 and GCN2 models. The model is optimized by minimizing the following boundary-based loss function:

$$L = \sum_{(e1,e2)\in\mathbb{L}} \sum_{(e1',e2')\in\mathbb{L}'} max\{0, f(X(e1), X(e2)) + \lambda - f(X(e1'), X(e2'))\} \tag{5}$$

Among them, $\mathbb{L}'$ is obtained by randomly replacing the entity of the seed entity pair; λ is the hyperparameter that separates the alignment of the positive and negative entities; $f(.)$ represents the distance between the calculation entities $e1$ and $e2$ vectors. SGD is used to minimize the loss function and obtain the optimal model.

Complexity analysis. Complexity $O(|E|d(l)d(l+1))$ for convolution operations on the knowledge graph, where $|E|$ is the number of edges of the knowledge graph, $d(l+1)$ is the feature vector dimension, and $d(l)$ is the number of convolution kernels.

4.3. TransE attribute information embedding

In this part, we use the parameter sharing (PS) method and TransE for the embedding of attribute triples. This part can also be achieved by other joint and knowledge embedding methods. We define this part's scoring function as:

$$A = P_A + K_A \tag{6}$$

Among them, P_A and K_A represent the scores of the PS joint model and the TransE attribute triples embedding model respectively.

4.3.1. Parameter sharing (PS) joint model

We must first learn the knowledge representation in different KGs. However, in order to achieve entity alignment, we must add the knowledge representations of different KGs to a unified semantic space. In this unified space, we can calculate the entities distance between different KG to find the aligned entity. We perform joint embedding on the alignment seed set to achieve this goal.

After research, the existing commonly used joint embedding models can be roughly divided into three types: translation-based models, linear transformation models and parameter sharing models.

Translation-based model. Inspired by knowledge representation learning methods (such as TransE) based on translation models, alignment can be regarded as a special relationship between entities, and a specific alignment translation operation can be performed between aligned entities to learn joint embedding. Formally, for a given two alignment entities $e1 \in E1$ and $e2 \in E2$, we assume that there is an alignment relationship $r^{(E1 \rightarrow E2)}$ such that $e1 + r^{(E1 \rightarrow E2)} \approx e2$. Therefore, the energy function of the joint embedding method is defined as: $E(e1, e2) = \parallel e1 + r^{(E1 \rightarrow E2)} - e2 \parallel$.

Linear transformation model. The linear transformation model is a learning method. By learning the linear transformation between the knowledge embeddings of different KGs, the transformation matrix is finally obtained. For a given two alignment entities $e1 \in E1$ and $e2 \in E2$, we define a transformation matrix $M^{(E1 \rightarrow E2)}$, so that $M^{(E1 \rightarrow E2)} e1 \approx e2$. Therefore, we define the energy function as: $E(e1, e2) = \parallel M^{(E1 \rightarrow E2)} e1 - e2 \parallel$.

For the translation-based model and the linear transformation model, we can define its score function as the sum of the energy functions of all the aligned seeds, which is formalized as: $J_{(T/L)} = \sum_{(e1,e2) \in \mathbb{L}} \alpha E(e1, e2)$, where α is a weighting factor.

Parameter sharing (PS) model. Since the aligned entities have the same meaning in different KGs, it is intuitive to make these aligned entities share the same embedding. This method is the parameter sharing method. Formally, for each aligned entity pair (e, e'), we define $e \equiv e'$. The parameter sharing (PS) model can intuitively and efficiently calibrate the entities in KG_1 and KG_2 into the same semantic space. In this model, no variables are involved, so the scoring function of the model is: $P = 0$.

The first two models regard the joint embedding of knowledge as the regularization of knowledge representation learning, which involves the integration of variables and variable transformations. So this will lead to embedding errors. The third method is simple, intuitive and effective, which directly regards the aligned entities as the same embedding, without introducing variables or embedding errors. Therefore we use the third method—parameter sharing model.

4.3.2. Attribute triples embedding

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

TransE. In TransE, we consider a in each triple (e, a, v) as a transformation from e to v. By constantly adjusting e, a, and v in the embedding space, we make $e + a \approx v$ as much as possible. Therefore its energy function is defined as:

$$E(e, a, v) = \parallel e + a - v \parallel \tag{7}$$

In addition, we use a margin-based scoring function as the training target, which is defined as:

$$K_A = \sum_{AT \in \{AT_1, AT_2\}} \sum_{(e,a,v) \in AT} L(e, a, v) \tag{8}$$

where $L(e, a, v)$ is a marginal loss function, which we define as:

$$L(e, a, v) = \sum_{(e',a',v') \in AT^-} [\gamma + E(e, a, v) - E(e', a', v')]_+ \tag{9}$$

where $[x]_+ = max\{0, x\}$ represents the largest number between 0 and x, γ is a marginal constant, and AT^- represents the negative sample set of the attribute triple set AT, which is defined as:

$$\begin{aligned} AT^- = \{(e', a, v) \mid e' \in E\} \cup \\ \{(e, a, v') \mid v' \in V\} \cup \{(e, a', v) \mid a' \in A\} \end{aligned} \tag{10}$$

where $(e, a, v) \in AT$ and AT^- is obtained by randomly replacing any element in the attribute triples (e, a, v).

4.4. Joint embedding strategy

In order to study the effect of combining the results of structure embedding and attribute information embedding on the performance of EA tasks, we design three joint methods for the embedding results of structure and attribute information, namely connection strategy, weight allocation strategy, and iteration embedding strategy.

We present the structural embedding results obtained in Section 3.2 as E_{RT}, and the embedding results of attribute information obtained in Section 3.3 as E_{AT}. E_{RT} is represented by two-part embedded sets of E_{RE} and E_{RR}, where E_{RE} represents the entity embedding results in the structure information embedding part, and E_{RR} represents the relation embedding result in the structure information embedding part. Similarly, E_{AT} is represented by three-part embedded sets of E_{AE}, E_{AA}, and E_{AV}, where E_{AE} represents the entity embedding result in the attribute information embedding part, E_{AA} represents the attribute embedding result in the attribute information embedding part, and E_{AV} represents the attribute value embedding result in the attribute information embedding part. The various joint strategies are described in detail below.

4.4.1. Embedding connection strategy

The connection strategy refers to the direct connection between the E_{RE} obtained in Section 3.2 and the E_{AE} obtained in Section 3.3 to form a complete entity embedding result E_E. That is, $E_E = [E_{RE}; E_{AE}]$, where ; indicates a connection operation.

4.4.2. Embedding weight allocation strategy

The weight allocation strategy refers to setting the weight of E_{RE} obtained in Section 3.2 and E_{AE} obtained in Section

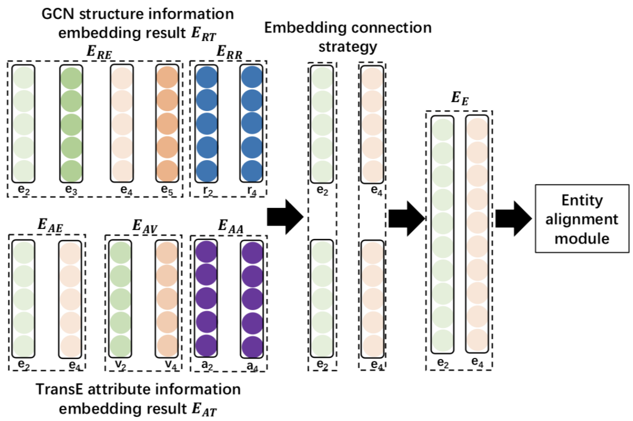


Figure 4: Connection strategy.

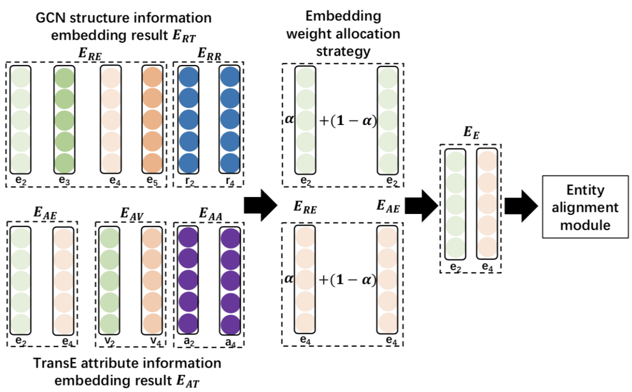


Figure 5: Weight allocation strategy.

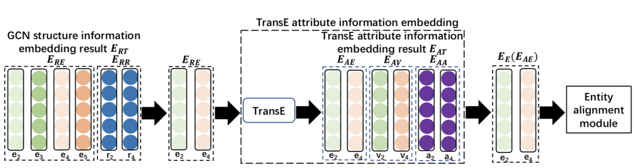


Figure 6: Iteration strategy.

3.3 respectively. The sum of weights is set to 1. After the weight is set, the entity embedding results of the two are added to form a complete entity embedding result E_E. That is, $E_E = \alpha E_{RE} + (1-\alpha)E_{AE}$, where α represents the weight assigned to E_{RE}, $(1-\alpha)$ represents the weight assigned to E_{AE}, and $\alpha \in (0, 1)$.

4.4.3. Embedding iteration strategy

The iteration strategy refers to the first choice to embed structural information (section 3.2) or the attribute information (section 3.3) (We will first take section 3.2 as an example). The structure information is obtained as E_{RE}, and the obtained E_{RE} is then input into TransE with attribute information used as the initialization entity embedding in attribute information embedding part. Afterwards, we obtain E_{AE}, which is also the final entity embedding result E_E, that is, $E_E = E_{AE}$.

4.4.4. Entity alignment

After the GCN structure information embedding, the TransE attribute information embedding and the embedding joint part, we obtain the final entity embedding result E_E. Because in the embedding semantic vector space, entities with similar meanings are often closer, we can perform EA based on the semantic distance between entities in the embedding semantic vector space.

We define the formula of semantic distance as $E(e_1, e_2) = \| e_1 - e_2 \|_{(L1/L2)}, \forall e_1 \in E_1, e_2 \in E_2$, where $L1/L2$ indicates that the distance is calculated using the L1 or the L2 normal form. Therefore, for an unaligned entity e_1 in one KG, we can find the entity $\hat{e_2}$ that is closest to it in another KG, that is, $\hat{e_2} = argmin_{e_2}(E(e_1, e_2))$.

5. Experiment

5.1. Baseline

We select MTransE[5], JAPE[14], GCN[18], and INGA[11] models as the baseline models. Among them, MTransE is a cross-language knowledge graph entity alignment model that only uses relational embedding, and GCN is an entity alignment model that further learns the complex structure information of the graph network; JAPE is a cross-language entity alignment model that retains attribute embedding; INGA is a preserving graph structure Entity alignment model of information and entity attribute information. These models will be compared with the ones proposed in this article.

5.2. Model Variants

To evaluate different components of our model, we provide two implementation variants for ablation studies, including (1)Our.SE: GCN structure information embedding module; (2)Our.AE: parameter sharing TransE model with initialization of normal distribution. These variants studies only consider the effect of structure embeddings or attribute embeddings for entity alignment.

5.3. Evaluation metrics

The EA task has the following evaluation indicators: the proportion of correct answers ranked in the top n (n can be 1, 10, 50, etc., namely Hits@1, Hits@10, Hits@50, etc.). In general, higher Hits@n indicates better alignment.

5.4. Datasets

The data set in this paper is from DBP15K, which was constructed by Sun et al. DBP15K is a cross-language KG constructed from DBPedia, and there are connections between entities in different languages of DBPedia (ie, interlanguage links, or ILLs). 15,000 ILLs from popular entities (degrees greater than 4) were extracted, and their relationships and attributes were combined in DBPedia to form DBP15K. DBP15K is available in Chinese, English, Japanese and French. Table 2 describes the details of the dataset. In the experiments, known aligned entities are used for model training and testing.

Table 2
Statistics for the datasets

DBP15K		Entities	Rel_Triples	Attr_Triples	ILLs
FR-EN	French	66858	192191	528665	15000
	English	105889	278590	576543	
JA-EN	Japanese	65744	164373	354619	15000
	English	95680	233319	497230	
ZH-EN	Chinese	66469	153929	379684	15000
	English	98125	237674	567755	

5.5. Experimental setup

5.5.1. *GCN structure information embedding settings*

We use Adam optimizer to optimize the GCN model. For the setting of hyperparameters: the learning rate is set to 0.1, the dropout is set to 0.5, and the λ in boundary-based loss function is set to 3. We randomly initialize the structural information embedding, including entity embedding and relation embedding, and the embedding dimension is set to 200. We set the number of iterations of the model to 600.

5.5.2. *TransE attribute information embedding settings*

For comparison, we refer to the experimental setting of the baseline and use Stochastic Gradient Descent (SGD) as the optimization method; the knowledge $E = \{e \mid e \in E\}$, $A = \{a \mid a \in A\}$, and $V = \{v \mid v \in V\}$ are all initialized to a normal distribution; all models are based on the same dimensions n=200 and epochs=3000, and the embedding of entities, attributes and attribute values use the same dimension n=200. For hyperparameters, we set $\gamma = 1.0$ and learning rate $\lambda = 0.001$.

5.5.3. *Joint embedding policy settings*

We use L1 normal form to calculate the semantic distance between entities.

5.6. Experimental results

Our experimental results are shown in Table 3. Our.CS represents the result of using the connection strategy, Our.WCS(α:$1 - \alpha$) represents the result of using the weight allocation strategy, and α represents the weight of the structural information entity embedding result. Our.IS represents the result of using the iteration strategy.

It can be seen from Table 3:

(1) The method we proposed has improved all the indicators of the physical alignment task. Compared with the baseline methods, our method improved the hits@1, hits@10, and hits@50 by 10.51%, 10.37%, and 7.57%, respectively, with the best experimental results under different methods. It is proved that after combining the attributes information obtained by TransE and the structural information learned by GCN, the entity embedding representation is more accurate and the entity alignment effect is improved.

(2) In the ablation experiments, Our.SE and Our.AE prove that a single model cannot adequately model multiple information. We design appropriate models to model multiple information and use multiple joint methods. Finally, the accuracy of the alignment task is improved.

(3) Among the three joint strategies, the experimental effect of the iterative embedding strategy is better than the other two strategies. In the experimental results of the three fusion strategies of embedding result connection strategy, embedding result weight distribution strategy and iterative embedding strategy, we can see that the experiment using iterative embedding strategy is better, which is in line with our expectations because iterative embedding strategy. It is equivalent to using TransE to embed attribute information on the basis of GCN-based structural information embedding results, and then adjusting the existing entity embedding results. Compared with the other two strategies, embedding alone and then combining the qualitative results The method has more opportunities for further adjustment, so its experimental effect is better.

(4) In the joint strategy of weight distribution, the size of relational data and attribute data in the experimental data set will affect the weight distribution and further affect the effect of the entity alignment task. In the experimental results using the weight distribution strategy of embedding results, the results of different weight distribution methods are different. It can be seen that as the proportion of embedding results in the embedding part of the GCN structure information increases, the experimental effect has a smaller downward trend, but this is because in our experimental data set, the amount of attribute information is twice greater than the amount of structural information, so the effect of the embedding of attribute information on the effect of the entity alignment task is a little larger than that of relational information. Relationship, in other experiments, the weights can be adjusted according to different data sets to obtain the best alignment effect.

6. Conclusions and future work

This paper combines the classic knowledge representation method TransE and the graph calculation method GCN for EA. Because GCN can make full use of the multi-step path of KG, so we use GCN to embed the structure information to preserve the relationship dependencies between entities. The TransE method is the most classic, high-performance and most commonly used method in knowledge representation learning methods, and attribute information does not contain complex structural information, it is only basic se-

Table 3
Experiment Result Table

	ZH-EN			EN-ZH		
	Hits@1	Hits@10	Hits@50	Hits@1	Hits@10	Hits@50
MTransE	30.83	61.41	79.12	24.78	52.42	70.45
JAPE	41.18	74.46	88.9	40.15	71.05	86.18
GCN	41.25	74.38	86.23	36.49	69.94	82.45
INGA	50.45	79.42	89.79	49.36	76.05	86.38
Our.SE	39.23	70.13	80.48	36.72	66.96	77.74
Our.AE						
Our.CS	60.38	89.04	96.19	59.54	86.15	92.69
Our.WCS(0.2:0.8)	56.82	82.59	89.91	55.92	82.22	89.36
Our.WCS(0.5:0.5)	56.07	82.31	87.99	55.35	81.53	88.56
Our.WCS(0.8:0.2)	54.45	80.28	85.75	53.48	78.83	84.18
Our.IS	62.36	90.69	97.24	61.03	88.91	95.97
	JA-EN			EN-JA		
	Hits@1	Hits@10	Hits@50	Hits@1	Hits@10	Hits@50
MTransE	27.86	57.45	75.94	23.72	49.92	67.93
JAPE	36.25	68.5	85.35	38.37	67.27	82.65
GCN	39.91	74.46	86.10	38.42	71.81	83.72
INGA	51.46	79.46	88.25	51.05	77.04	86.27
Our.SE	39.63	75.85	86.21	38.26	73.24	84.66
Our.AE						
Our.CS	59.56	87.41	93.93	58.43	83.87	90.74
Our.WCS(0.2:0.8)	56.14	81.39	87.18	55.17	78.30	85.05
Our.WCS(0.5:0.5)	55.37	81.62	88.22	54.77	78.85	85.86
Our.WCS(0.8:0.2)	54.76	80.57	86.10	54.30	77.30	83.06
Our.IS	61.33	90.14	96.75	60.31	85.94	93.50
	FR-EN			EN-FR		
	Hits@1	Hits@10	Hits@50	Hits@1	Hits@10	Hits@50
MTransE	24.41	55.55	74.41	21.26	50.6	69.93
JAPE	32.39	66.68	83.19	32.97	65.91	82.38
GCN	37.29	74.49	86.73	36.77	73.06	86.39
INGA	50.45	79.42	87.79	49.36	76.05	86.48
Our.SE	40.50	73.26	82.90	38.08	69.15	78.99
Our.AE						
Our.CS	58.77	86.73	94.08	57.56	83.14	89.59
Our.WCS(0.2:0.8)	56.27	82.92	88.70	55.70	81.23	88.48
Our.WCS(0.5:0.5)	55.98	81.17	87.69	55.14	79.64	85.83
Our.WCS(0.8:0.2)	55.28	80.10	85.69	54.51	79.88	85.45
Our.IS	60.45	87.55	94.11	59.69	86.40	92.79

quence information. Therefore we use TransE to embed attribute information. In addition, we propose three joint embedding strategies, namely connection strategy, weight allocation strategy, and iterative strategy. Experimental results show that our methods have improved the metrics of hits@1, hits@10 and hits@50 by an average of 10.51%, 10.37%, and 7.57% respectively compared with the baseline methods. The source code for this paper is available from `https://github.com/ChengRui536/TransE-GCN`.

In the future, we will explore the following research directions: (1) the embedding of entity descriptive information in KG for EA; (2) dual graph embedding; and (3) heterogeneous graph embedding.

CRediT authorship contribution statement

Haihong E: Conceptualization of this study, Methodology. **Rui Cheng:** Conceptualization of this study, Methodology, Software, Writing - Original draft preparation. **Jingru Liang:** Conceptualization of this study, Methodology, Software, Data curation. **Meina Song:** Conceptualization of this study, Methodology.

References

[1] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J., 2008. Freebase: A collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, ACM, New York, NY, USA. pp. 1247–1250. URL: `http://doi.acm.org/10.1145/1376616.1376746`, doi:`10.1145/1376616.1376746`.

[2] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J.,

Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data, in: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 26. Curran Associates, Inc., pp. 2787–2795. URL: http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf.

[3] Cao, Y., Liu, Z., Li, C., Liu, Z., Li, J., Chua, T.S., 2019. Multi-channel graph neural network for entity alignment, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy. pp. 1452–1461. URL: https://www.aclweb.org/anthology/P19-1140, doi:10.18653/v1/P19-1140.

[4] Chen, M., Tian, Y., Chang, K.W., Skiena, S., Zaniolo, C., 2018. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, AAAI Press. pp. 3998–4004. URL: http://dl.acm.org/citation.cfm?id=3304222.3304326.

[5] Chen, M., Tian, Y., Yang, M., Zaniolo, C., 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, AAAI Press. pp. 1511–1517. URL: http://dl.acm.org/citation.cfm?id=3172077.3172097.

[6] Guo, L., Sun, Z., Hu, W., 2019. Learning to exploit long-term relational dependencies in knowledge graphs. arXiv:1905.04914.

[7] Hao, Y., Zhang, Y., He, S., Liu, K., Zhao, J., 2016. A joint embedding method for entity alignment of knowledge bases, in: Chen, H., Ji, H., Sun, L., Wang, H., Qian, T., Ruan, T. (Eds.), Knowledge Graph and Semantic Computing: Semantic, Knowledge, and Linked Big Data, Springer Singapore, Singapore. pp. 3–14.

[8] Ji, G., He, S., Xu, L., Liu, K., Zhao, J., 2015. Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China. pp. 687–696. URL: https://www.aclweb.org/anthology/P15-1067, doi:10.3115/v1/P15-1067.

[9] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C., 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web 6, 167–195. URL: https://madoc.bib.uni-mannheim.de/37476/, doi:10.3233/SW-140134.

[10] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X., 2015. Learning entity and relation embeddings for knowledge graph completion. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571/9523.

[11] Pang, N., Zeng, W., Tang, J., Tan, Z., Zhao, X., 2019. Iterative entity alignment with improved neural attribute embedding, in: DL4KG@ESWC.

[12] Pei, S., Yu, L., Hoehndorf, R., Zhang, X., 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference, in: The World Wide Web Conference, ACM, New York, NY, USA. pp. 3130–3136. URL: http://doi.acm.org/10.1145/3308558.3313646, doi:10.1145/3308558.3313646.

[13] Qu, M., Tang, J., Bengio, Y., 2019. Weakly-supervised knowledge graph alignment with adversarial learning. arXiv:1907.03179.

[14] Sun, Z., Hu, W., Li, C., 2017. Cross-lingual entity alignment via joint attribute-preserving embedding, in: d'Amato, C., Fernandez, M., Tamma, V., Lecue, F., Cudré-Mauroux, P., Sequeda, J., Lange, C., Heflin, J. (Eds.), The Semantic Web – ISWC 2017, Springer International Publishing, Cham. pp. 628–644.

[15] Sun, Z., Hu, W., Zhang, Q., Qu, Y., 2018. Bootstrapping entity alignment with knowledge graph embedding, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization. pp. 4396–4402. URL: https://doi.org/10.24963/ijcai.2018/611, doi:10.24963/ijcai.2018/611.

[16] Trisedya, B., Qi, J., 2019. Entity alignment between knowledge graphs using attribute embeddings. Proceedings of the AAAI Conference on Artificial Intelligence 33, 297–304. doi:10.1609/aaai.v33i01.3301297.

[17] Vrandečić, D., Krötzsch, M., 2014. Wikidata: A free collaborative knowledge base. Communications of the ACM 57, 78–85. URL: http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext.

[18] Wang, Z., Lv, Q., Lan, X., Zhang, Y., 2018. Cross-lingual knowledge graph alignment via graph convolutional networks, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium. pp. 349–357. URL: https://www.aclweb.org/anthology/D18-1032, doi:10.18653/v1/D18-1032.

[19] Wang, Z., Zhang, J., Feng, J., Chen, Z., 2014. Knowledge graph embedding by translating on hyperplanes. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531/8546.

[20] Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., Zhao, D., 2019. Relation-aware entity alignment for heterogeneous knowledge graphs, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization. pp. 5278–5284. URL: https://doi.org/10.24963/ijcai.2019/733, doi:10.24963/ijcai.2019/733.

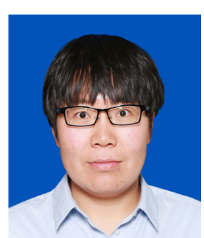
[21] Xiao, H., Huang, M., Hao, Y., Zhu, X., 2015. Transa: An adaptive approach for knowledge graph embedding. arXiv:1509.05490.

[22] Yang, H.W., Zou, Y., Shi, P., Lu, W., Lin, J., Sun, X., 2019. Aligning cross-lingual entities with multi-aspect information. arXiv:1910.06575.

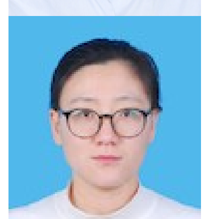
[23] Ye, R., Li, X., Fang, Y., Zang, H., Wang, M., 2019. A vectorized relational graph convolutional network for multi-relational network alignment, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization. pp. 4135–4141. URL: https://doi.org/10.24963/ijcai.2019/574, doi:10.24963/ijcai.2019/574.

[24] Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., Qu, Y., 2019. Multi-view knowledge graph embedding for entity alignment. arXiv:1906.02390.

[25] Zhu, H., Xie, R., Liu, Z., Sun, M., 2017. Iterative entity alignment via joint knowledge embeddings, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 4258–4264. URL: https://doi.org/10.24963/ijcai.2017/595, doi:10.24963/ijcai.2017/595.



Haihong E (1982-), female (Han), native of Liaoning Province, China. associate professor and Ph.D., School of Computer Science, Beijing University of Posts and Telecommunications, major research areas include deep learning, knowledge graph, natural language processing, big data and artificial intelligence



Rui Cheng (1995-), female (Han), native of Shanxi Province, China. Beijing University of Posts and Telecommunications, School of Computer Science, graduate student, postgraduate, the main research areas are KG and natural language processing.

Jingru Liang (1997-), female (Han), Chinese. Beijing University of Posts and Telecommunications, School of Computer Science, graduate student, postgraduate, the main research areas are KG and natural language processing.

Meina Song (1974-), female (Han), native of Shandong Province, China. professor and Ph.D., School

of Computer Science, Beijing University of Posts and Telecommunications, major research areas include medical KG, artificial intelligence and data mining.