





Knowledge Graph Representation Learning based on Entity Semantic Distance Classification

Yi Zhang^{1,2} , Wanhua Cao^{1,2} , Juntao Liu² , Yuanbin Wang²  and Ziyun Rao² 

¹□ College of Computer Science and Technology, Harbin Engineering University, Harbin, China, yzhang85@hrbeu.edu.cn, clarkmss@163.com

²□ Wuhan Digital Engineering Research Institute, Wuhan, China, prolay@163.com, 56781493@qq.com, rzy181234@163.com

Corresponding author: Yi Zhang, yzhang85@hrbeu.edu.cn

Abstract. In recent years, significant progress has been made in knowledge graph representation learning, which has shown promising results in knowledge computing applications such as relation extraction and knowledge reasoning. However, the unbalanced distribution of relations and entities in knowledge graphs leads to low training efficiency. To address this issue, this paper proposes a novel knowledge representation learning method based on entity distance classification. This method classifies entities based on their semantic distance on a specific relational plane, and employs different training strategies to increase the training opportunities for entities with low semantic distance differentiation. Moreover, the loss function is adjusted by introducing different residual weights, which allows for the allocation of different training opportunities to negative samples. The effectiveness of our approach is demonstrated by comparing it with mainstream knowledge representation models on various benchmark datasets.

Keywords: Knowledge representation learning, entity semantic distance classification, knowledge computing, semantic reasoning

DOI: <https://doi.org/10.14733/cadaps.2024.S7.256-269>

1 INTRODUCTION

In recent years, significant progress has been made in the theory and key technologies of knowledge computing, resulting in the construction of numerous knowledge graphs, such as YAGO [1], Freebase [2], DBpedia [3], Wikidata [4], NELL [5] and other general knowledge graphs of encyclopedias. Additionally, there are industry-specific knowledge graphs, such as the medical knowledge graph CEMRs [6]. Knowledge graphs offer a novel way of perceiving the real world. They usually organize knowledge in the form of the Semantic Web, where each node represents entities (countries, names, drugs, concepts, etc.), and each edge represents the relationship between entities (kinship, inclusion relation, etc.). This form describes the structured relationship fact between concrete entities and abstract concepts [7]. Therefore, most knowledge can often be represented as a triple (head entity, relation, tail entity), which corresponds to an edge in the

knowledge graph network and its two connected entities. In knowledge graph $G = (E, R, S)$, E is the set of entities, R is the set of relations. $S \subseteq E \times R \times E$ is the set of fact triplets (h, r, t) , where h , t , r denote head entity, tail entity and relationship, respectively. For example, the triplet (diuresis, symptom of, glycuressis) indicates a "symptom of" relationship between "diuresis" and "glycuressis".

Currently, knowledge graphs play an important role in many artificial intelligence tasks, such as entity recognition [8], semantic parsing, text classification, document summarization, subject indexing, intelligent recommendation [9][10], information extraction [11], and knowledge question answering [12].

The application of knowledge graphs still faces significant challenges, including data sparsity and computational efficiency [7]. However, the advancement of deep learning has led to the exploration of knowledge graph representation learning, which demonstrates a strong ability to represent knowledge in various applications such as relation extraction and knowledge reasoning. Knowledge representation learning (KRL) usually projects the semantic structure information of entities and relations into a low-dimensional vector space [13]. Compared with traditional representation methods, knowledge representation learning can represent the entities and relations more densely, thus reducing the computational complexity in application. In addition, the performance of knowledge acquisition, fusion and inference can be improved by effectively measuring the semantic relevance of entities and relationships in knowledge graphs. Based on the above advantages, knowledge representation learning develops vigorously in the application of knowledge graphs, and knowledge representation models keep emerging, such as TransE [14] and its derivative models [15][16], ComplEx [17], DistMult [18] and RotatE [19].

Although these models show their advantages and innovations in some aspects, they seldom consider the unbalanced distribution of entities under different relational semantic planes. Although these models demonstrate their advantages and innovations in some aspects, they often neglect the imbalanced distribution of entities across different semantic planes. During training, negative sample triples are constructed using substitution methods such as uniform random sampling of head and tail entities (unif) [14] or Bernoulli random sampling (bern) [15]. However, these methods do not consider the semantic distance between the substituted entities on a specific relational plane. For instance, the entities "male" and "female" have a high correlation with the relation "gender," and therefore, their semantic distance is close on the relation plane of "gender". When negative sample construction is performed on a positive triple (Diego Armando Maradona, gender, male), it may produce a negative triple (Diego Armando Maradona, gender, female), which is a high contribution rate negative sample. While the entity "Cardiopathy" and "glycuressis" are very relevant to the entity "Diego Armando Maradona" (they were the cause of death in Diego Armando Maradona), they are less relevant to the relation "gender". Therefore, the semantic distance in the "gender" relation plane is far away. The negative samples constructed from them, such as (Diego Armando Maradona, gender, Cardiona) and (Diego Armando Maradona, gender, glycuressis), are of low value. If these samples are extensively used for training knowledge representation learning models, it may reduce the training effect and learning performance of the model.

To address the problem of low training efficiency caused by unbalanced distribution of entities in knowledge graphs, this paper proposes a knowledge representation learning method based on entity distance. Each relation in training samples is classified according to the affinity of semantic distance between entities on the relation plane. Different training strategies are adopted: entities with affinity (i.e., close semantic distance, with dense distribution, small differentiation) are trained more, while entities with sparsity (i.e., semantic distance is far away, sparse distribution and large differentiation) are trained less. This improves the training efficiency of the knowledge representation model. Additionally, the loss function of model training is optimized, and different residual weights are assigned to different entities to further enhance the predictive ability of the knowledge representation learning model. The effectiveness of this knowledge graph representation learning method is verified on general datasets.

The remainder of this paper is organized as follows. The next section introduces four scopes of knowledge representation learning, including vector representation space, score function definition, model training strategies, and negative sampling strategies; the section 3 elaborates on our knowledge representation learning method based on entity distance classification; the experimental part in section 4 compares the performance of our knowledge representation model with mainstream models, and verifies the validity of our method. Finally, we summarize our work.

2 RELATED WORK

Knowledge representation learning is a key area of research for knowledge graphs, with the goal of embedding knowledge into a low-dimensional continuous semantic space. Many models have been proposed, with a focus on vector representation space, score function definition, model training strategies, and negative sampling strategies.

2.1 Vector Representation Space

The real-valued vector space is widely used as a representation space for embedding entities and relationships. It allows entities and relations to be encoded as vectors or matrices and can capture relational interactions. TransE considers the relation r as a transfer vector between the head entity vector h and the tail entity vector t for each triple and assumes $h + r \approx t$ holds. However, TransE has limitations in representing one-to-many and many-to-one relationships. To address these limitations, TransH introduces a hyperplane that projects the head and tail entities of a triple onto a relationship-specific hyperplane. TransR [16] further extends this idea by introducing a space for the separation of entities and relationships. Entities are projected into the relationship space using a projection matrix. Similar representation spaces are also used in other translation models, such as TransD [20], while semantic matching models prefer to employ pure vector spaces [21], and relational projection matrices [22].

Expanding from a real-valued vector space to a complex vector space enables the representation of entities and relationships with richer representational capabilities. ComplEx uses the Hamiltonian product method to model symmetric and antisymmetric relationships. QuatE [23] extends the complex space to a quaternion space and combine head entities and relations by quaternion multiplication.

KG2E [24] and TransG [25] use a Gaussian distribution to represent entities and a mixed Gaussian distribution to represent relational embedding. There are other representation spaces, such as the manifold represented by ManifoldE [26], and the group space represented by TorusE [27].

2.2 Score Function Definition

The score function is used to measure the reliability of a triple, also known as the energy function, and is the foundation of an energy-based learning framework. The scoring function is usually divided into distance-based and semantic matching-based scoring function.

The distance-based score function measures the semantic distance between two entities. The original SE model [28] utilized two mapping matrices and the L1 parametrization to learn the embedding representation of entities and relations. Then, variants and extended versions of the TransE model have been proposed. TransD constructs a dynamic mapping matrix. TransR projects the head entity and the tail entity from the entity space to the space of relations through a projection matrix. TransA [29] model replaces the Euclidean distance with the Mahalanobis distance to achieve adaptive metric learning.

The semantic matching-based scoring function calculates the semantic similarity. DistMult [18] devises a simplified bilinear form by restricting the relational matrix to a diagonal matrix for multi-relational representation learning. HoLE [269] employs embedded cyclic correlation to learn the combined representation and capture interaction information in relational data. ANALOGY [269]

focuses on the inference of multiple relations, modeling the analogical structure of relational data. SME [30] calculates the semantic match between entity-relationship pairs. Finally, CrossE [31] introduces crossover interactions, using an interaction matrix to model the two-way interaction of entities and relationships.

2.3 Model Training Strategies

There are two typical training strategies in KRL, which are based on the margin-based loss [14] and based on the logistic loss [17].

The margin-based loss defines the loss function as the training criterion:

$$L(\theta) = \sum_{\delta^+ \in S^+} \sum_{\delta^- \in S^-} \max(0, f_r(h, t) + \gamma - f_r(h', t')) \quad (2.1)$$

where θ denotes all parameters in the KRL model, $\delta^+ = h, r, t$ is a positive sample in positive sample set S^+ , $\delta^- = h', r', t'$ is a negative one in negative sample set S^- , γ is a margin separating them.

The logistic loss define the loss function based on logistic regression:

$$L(\theta) = \sum_{\delta^+ \in S^+} \log(1 + \exp(-y_r(h, t))) + \sum_{\delta^- \in S^-} \log(1 + \exp(-y_r(h', t'))) \quad (2.2)$$

where $y_r(h, t)$ denotes the energy of the triple h, r, t , which can be defined as follows:

$$y_r(h, t) = -f_r(h, t) + c \quad (2.3)$$

where c is a bias constant.

Some knowledge representation models, such as ConvE [32], RotatE [269], etc. also use other types of loss functions. For all these types of loss functions, specific regularizations can also be applied, such as L1 (or L2) on parameters or constraints. And all can be easily optimized for these loss functions by SGD, Adagrad, etc.

2.4 Negative Sampling Strategies

To use margin-based loss in knowledge representation learning, it is necessary to generate negative triples because only correct triples are included in it. Various methods have been proposed for generating negative samples. However, generating negative instance triples by uniform substitution may lead to invalid negative samples. TransH addresses this issue by assigning different weights to head and tail entity substitutions based on association features when generating negative instance triples. For instance, when replacing one side of relation 1 to n , it tends to replace one side of 1 instead of n to reduce the probability of generating an invalid negative instance triple. Negative samples for any triple of a specific relation can be generated by replacing the head entity with a certain probability and the tail entity with another probability.

The quality of the generated negative samples is also crucial, and several adversarial learning techniques are introduced for negative sampling. For instance, KBGAN [33] uses a probability-based log-loss embedding model for its generator, while RotatE proposes self-adversarial negative sampling based on its scoring function.

Trouillon [17] suggest that generating more negative samples generally results in better predictive results, and a good compromise between accuracy and training time is to have 50 negative samples per positive sample.

3 OUR METHOD

In this paper, we propose a knowledge representation learning method based on entity distance, which consists of five steps: (1) Initializing the vector representation of entities and relationships

in the training sample of the knowledge graph; (2) Classifying all entities according to their semantic distances for each relationship semantic plane in the training sample; (3) Generating negative samples using different strategies based on the entity semantic distance classification in step 2; (4) Optimizing the loss function of training by introducing residual weights; (5) Substituting triples in knowledge graph datasets into the model for training and solving the representation vectors of entities and relations. The algorithm flow is shown in Figure 1.

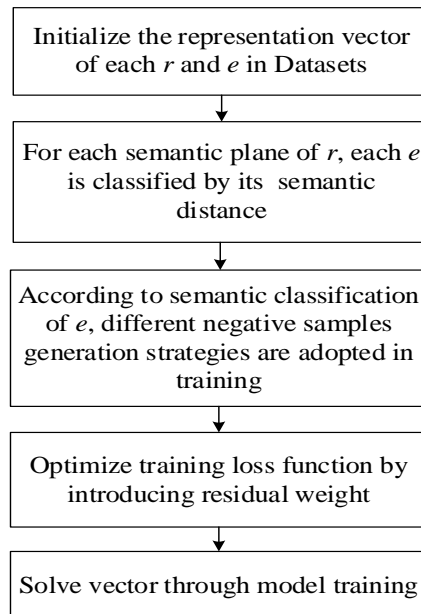


Figure 1: Flow chart of knowledge representation learning method based on entity semantic distance classification.

The method improves the set used to construct negative samples by the distance difference between entities, which makes the construction of negative samples more efficient. More training opportunities are allocated to entities that are closer, and different weights are set in the loss function for different affinities to improve the training efficiency of knowledge representation learning. The proposed method is described in detail below.

3.1 Entity and Relationship Initialization

Firstly, it is necessary to vectorize the relationships and entities in the training sample data of the knowledge graph. Let the entity set be $E = e_1, e_2, \dots, e_n$, where e_i is representing an entity in the entity set E . Similarly, there is the set of relations $R = r_1, r_2, \dots, r_m$, where r_i represents a relation in the set of relations R . Thus, the knowledge graph is $S_{h,r,t} = h, r, t, h, t \in E, r \in R$. Depending on the selected representation space, each element in the entity set E and the relationship set R can be vectorized.

3.2 Entity Classification based on Semantic Distance

For each relationship $r \in R$ in the training sample data, i.e., the knowledge graph $S_{h,r,t}$ (which can be abbreviated as the set of triples S), search the knowledge graph to find all triples that contain the relationship r . Put the head and tail entities that appear in these triples into set S' , and put the entities that do not appear in these triples into set S'' .

Specifically, for each relationship $r \in R$, scan the knowledge graph to find all triples h,r,t that contain the relationship r , and put the head entity h and tail entity t of these triples into set E_r' . Then, E_r'' can be found $E_r'' = E - E_r'$.

For example, for the triple h,r,t in the knowledge graph $S_{h,r,t}$ (which can be abbreviated as the set of triples S), the vector corresponding to the head entity h ($h \in E$, italicized by h) is \mathbf{h} (bolded \mathbf{h} denotes the vector), the vector corresponding to the relationship r ($r \in R$, italicized by r) is \mathbf{r} (bolded \mathbf{r} denotes the vector), and the vector corresponding to the tail entity t ($t \in E$, italicized by t) is \mathbf{t} (bolded \mathbf{t} denotes the vector).

For each relation $r \in R$, the set of entities E is divided into E_r' and E_r'' . The entities in the entity set E_r' appear in the triples of relations r in knowledge graph, i.e., $E_r' = \{e \mid e \in E, h, t \in E, h, r, e \in S \cup e, r, t \in S\}$. E_r'' is the complementary set of E_r' , i.e., $E_r'' = E - E_r'$.

3.3 Negative Sample Construction Based on Semantic Distance Classification

In the early negative sample construction methods, for the correct triple h,r,t in the training sample data, the head entity h or tail entity t is replaced by a random entity with a new head entity h' or tail entity t' , where h' and t' are also belonged to the set of entities E , but it did not distinguish whether the replacement entity (h' or t') ever appeared at this location.

The improvement idea of this method in negative sample construction is to provide more opportunities to select entities in set E_r' for constructing negative samples in each training epoch, and less opportunities to select entities in set E_r'' . This idea comes from the training of students to do multiple-choice questions in the real world. For example, the question "What was the cause of Maradona's death?", if the answer options are set to "Cardiopathy, Penicillin, Metronidazole, Norfloxacin", the student can easily select the correct answer "Cardiopathy". However, if the answer options are set to "Cardiopathy, HIV, COVID-19, Leukemia" (all of which were possible fatal diseases), it will obviously be more difficult for students to choose the correct answer. This is because the semantic distance between "Cardiopathy" and "Penicillin", "Metronidazole", and "Norfloxacin" will be farther, while the semantic distance between "Cardiopathy" and "HIV", "COVID-19", and "Leukemia" will be closer. Therefore, training on positive and negative samples with closer semantic distances (confusing options and easy-to-make-mistakes multiple-choice questions) can improve the accuracy of semantic reasoning through semantic distance calculation in subsequent stages.

The steps to implement the improvement idea for negative sample construction are as follows:

1) Given the sets E_r' and E_r'' , assign a probability ω (ω is a real number and $\omega \in [0,1]$) to each of them. Take a random value $pr \in [0,1]$, if $pr \in [0,\omega]$, select an entity from the set E_r' to

construct a negative sample; otherwise, select an entity from the set E_r'' to construct a negative sample.

2) After selecting an entity, replace the head or tail entity in the original triple using Bernoulli random sampling to construct a negative sample, and ensure that the constructed negative triple does not exist in the knowledge graph.

3) For each triple in the knowledge graph S containing the relation r , the following two quantities are counted:

a) The average number of tail entities corresponding to each head entity, denoted as $tail_per_head$;

b) The average number of head entities corresponding to each tail entity, denoted as $head_per_tail$.

Thus, the probability ε of replacing the head entity can be calculated as

$$\varepsilon = \frac{tail_per_head}{tail_per_head + head_per_tail} \quad (3.1)$$

4) Take a random number $pr1 \in [0,1]$. If $pr1 \in [0,\varepsilon]$, replace the head entity h in the original triple h,r,t to generate a negative sample h',r,t , and ensure that h',r,t is not in the knowledge graph S ; otherwise, replace the tail entity t in the triple h,r,t to generate a negative sample h,r,t' and ensure that h,r,t' is not in the knowledge graph S .

3.4 Loss Function Optimization

In this section, the loss function is optimized based on the difference in entity distance to improve training efficiency.

The Margin Loss Function is a commonly used loss function defined based on SVM (Support Vector Machine). It can establish a gap to separate positive and negative samples, where γ represents the gap distance. In each training iteration, it stretches positive and negative samples based on the gap by computing the loss function.

Our method improves the Margin Loss Function by introducing different stretch rate weights. Two residual weights, α and β , are introduced to produce different stretching effects on the distance between positive and negative samples generated from two entity sets E_r' and E_r'' during model training. The aim is to assign different stretching rates to negative samples that are easily confused when constructed from E_r' , because they have more training opportunities, so the stretching rate is lower; while negative samples that are easily distinguished when constructed from E_r'' , because they have fewer training opportunities, so the stretching rate is higher. Setting different stretching weights is also a way to allocate different balances to negative instance construction opportunities.

The improved loss function is defined as:

$$\begin{aligned}
L = & \sum_{h,r,t \in S_{h,r,t}} \sum_{h',r',t' \in S'_{h,r,t}} \left[\alpha f_r(h,t) + \gamma - f_r(h',t') \right]_+ \\
& + \sum_{h,r,t \in S_{h,r,t}} \sum_{h'',r'',t'' \in S''_{h,r,t}} \left[\beta f_r(h,t) + \gamma - f_r(h'',t'') \right]_+ \\
& + \eta \left(\sum_{e \in E} \left[\|e\|_2^2 - 1 \right]_+ + \sum_{r \in R} \left[\|r\|_2^2 - 1 \right]_+ \right)
\end{aligned} \tag{3.2}$$

where the operator $[x]_+ = \max(0, x)$, the operator $\|x\|_2^2$ denotes the L2 distance squared of the vector x , $S_{h,r,t}$ denotes the set of triples existed in the knowledge graph. $S'_{h,r,t}$ denotes the set of negative sample triples generated by selecting and replacing entity in E'_r , and $S''_{h,r,t}$ denotes the set of negative sample triples generated by selecting and replacing entity in E''_r .

$f_r(h, t)$ indicates the value of the scoring function of the knowledge representation model (If the TransE is introduced into our method, $f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{\ell_{1/2}}$; If the RotatE is applied, $f_r(h, t) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|_{\ell_{1/2}}$ and \circ is Hadamard products). $f_r(h', t')$ and $f_r(h'', t'')$ indicate the scoring functions for constructing negative triples of entities from the sets E'_r and E''_r , respectively. \mathbf{w}_r is the standard vector of the hyper plane of a particular relation. \mathbf{w}_r^\top is the transpose of the vector \mathbf{w}_r . The operation symbol $\|\cdot\|_{\ell_{1/2}}$ indicates the L1 or L2 distance of the calculated vector. α and β denote the weight of the scoring function for constructing negative triples from the set of S' and S'' , respectively. γ is a constant to separate positive and negative samples. The tail term $\eta \left(\sum_{e \in E} \left[\|e\|_2^2 - 1 \right]_+ + \sum_{r \in R} \left[\|r\|_2^2 - 1 \right]_+ \right)$ is used to prevent over-fitting during the training process.

Then, the improved model can be optimized using SGD, Adagrad, Adam, etc. All the triples in the knowledge graph are trained according to the loss function and the loss function is optimized based on the difference in entity distances. Finally, representation vectors of all entities and relations are obtained.

4 EXPERIMENT

In this section, we validate the proposed knowledge representation learning method through link prediction task on multiple benchmark datasets. We also investigate the impact of parameter changes in the proposed model on the performance of the knowledge representation. First, commonly used knowledge graph benchmark datasets are introduced. Then, the evaluation protocol and implementation are described in the experimental settings. By performing a comparative analysis of the experimental results, we demonstrate that the proposed approach based on entity semantic distance classification is successful in enhancing performance.

4.1 Datasets

The datasets commonly used in knowledge representation learning research include Freebase [2], WordNet [34], YAGO [1], and others. In order to conduct objective experimental comparisons with a wider range of representation models, we use five widely chosen datasets, namely FB15k,

WN18, FB15k-237 [35], WN18RR, and YAGO3-10 [32], to evaluate our method, as detailed in Table 1.

FB15k, WN18, and YAGO3 are subsets extracted from Freebase, WordNet, and YAGO, respectively. They are constructed as performance benchmarks for knowledge representation learning. WN18 and FB15k have test set leaks due to the presence of inverse relations. Knowledge representation learning employs a simple rule-based model, and state-of-the-art results can be obtained as well. As a subset of FB15k, the inverse relation of FB15k-237 is removed. Similarly, as a subset of WN18, the inverse relations in WN18RR [32] have also been removed. Therefore, the key to link prediction in FB15k and WN18 is modeling and inferring symmetric/asymmetric and inversion patterns, while the key to link prediction in FB15k-237 and WN18RR can be attributed to modeling and inferring their symmetry/asymmetry and composition patterns.

<i>Dataset</i>	<i>#Ent</i>	<i>#Rel</i>	<i>#Train</i>	<i>#Valid</i>	<i>#Test</i>
FB15k	14,951	1,345	483,142	50,000	59,071
WN18	40,493	18	141,442	5,000	5,000
FB15K-237	14,541	237	272,115	17,535	20,466
WN18RR	40,493	11	86,835	3,034	3,134
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Table 1: Datasets used in the experiment.

4.2 Experiment Settings

We traverse all the triples in the test set and replace the head entity of each triple with all other entities in the entity set one by one. These constructed triples are then scored by the score function defined by the model and sorted in ascending order. So we obtain the rank of the correct entity. The process is then repeated by replacing the tail entity. We calculate the proportion of correct entities ranked in the top N based on this procedure. Considering that the invalid negative triples constructed by the above method may be in the training and validation sets, they may rank above the test triples and affect the experimental results. These triples should not be considered as errors, because they are correct along with the original triples. Therefore, we delete those constructed invalid negative triples that appear in the training, validation, or test sets to ensure that they do not affect the final experimental results. In this setup, we calculate the proportion of correct entities ranked in the top 10/1 (Hits@10, Hits@1). We also calculate the mean rank (MR) and mean reciprocal rank (MRR) of the test triples in this setup.

The parameters that need to be set in the experiment include: the dimension of the vector m , the batch size, the training epochs and the margin hyperparameter γ .

Additionally, for our method, we need to experimentally determine the assignment probability ω of sets E_r' and E_r'' , where ω are real numbers, and $\omega \in [0,1]$. When calculating the loss function, α and β denote the weight of the scoring function when selecting triples from S' and S'' sets to construct negative samples, respectively.

The dimensions m are in the range {100, 200, 512, 1024}. The training rounds are in the range {1000, 3000, 6000}. The batch size is in the range {100, 200, 400}. The assignment probability ω is in the range {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}. The interval parameter γ is in the range {1.0, 2.0, 4.0, 8.0}. The parameter pair α, β is in the range {(1, 1), (1, 3), (3, 1), (1, 5), (5, 1), (1, 7), (7, 1), (1, 10), (1, 1)}.

4.3 Results

In this paper, TransE and RotatE models are introduced into our method, and the improved models are DC-TransE and DC-RotatE models.

(1) Study the impact of the model parameters ω and α, β on prediction performance.

As shown in Figure 2, we explore the performance changes (Hits@10) of the DC-TransE and DC-RotatE models with varying values of the parameter ω within the range of $[0, 1.0]$. When ω is between $[0, 0.1]$, the predictive ability of the model shows an upward trend, while performance gradually declines for ω in the range $(0.1, 1]$. Although 0.1 is the smallest value of ω , it should be considered in real-world datasets because they are often sparse and have a long-tail distribution, such as FB15k. The proportion of the number of entities associated with relationships (i.e., the number of entities in E_r') to the total number of entities is less than 1%. For example, in FB15k, among 1345 relationships, only 279 relationships have a proportion of the number of entities associated with them and the total number of entities greater than 1%, which accounts for only 21% of the total number of relationships. Therefore, 0.1 substantially increases the training opportunities for entities in E_r' . The experimental results validate our viewpoint: by increasing the training opportunities for entities with low discriminability of semantic distance, the link prediction accuracy of the knowledge representation learning model can be effectively improved. However, excessive increases may also reduce the link prediction performance. For example, when increasing ω to 0.7 and 0.9, the prediction performance decreases significantly. This indicates that although improving the model's discriminative ability for entities with similar semantic distances, it may lose its discriminative ability for entities with distant semantic distances. For example, when predicting the tail entity of the triple (Diego Armando Maradona, cause of death, ?), the model's discriminative ability is enhanced for entities such as "Cardiopathy", "glycuresis" and "diuresis" due to increased training opportunities. However, its discriminative ability is weakened for entities that have not appeared or been trained, like "male". Moreover, such entities with distant semantic distances may exist in large numbers, which is also due to the sparsity of knowledge graphs and the fact that the number of entities in E_r'' is often greater than E_r' .

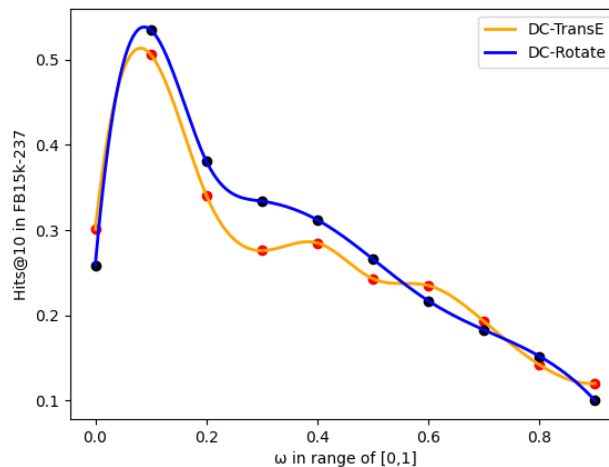


Figure 2: The link prediction performance changes (Hits@10) of the DC-TransE and DC-Rotate models within the range of $[0,1.0]$ for the parameter ω .

Furthermore, through experiments, we find that when $\alpha < \beta$, good results can be achieved, particularly with $\alpha, \beta = 1, 5$. This validates our previous viewpoint: for negative samples that are easily confused and constructed from E_r' , lower stretching rates can achieve better performance because they have more training opportunities; for negative samples that are easy to identify, higher stretching rates are more appropriate because they have fewer training opportunities. The tuning optimization of the parameters α and β essentially balances the training opportunities for different entities adjusted by ω .

(2) In Table 2, the performance of DC-TransE and TransE are compared on WN18 and FB15K. In Table 3, the performance of DC-RotatE and mainstream models are compared on WN18RR, FB15K237 and YAGO3-10. The experimental results demonstrate that our improved approach, which is based on entity distance, can effectively enhance the link prediction ability of the knowledge representation model by increasing the training opportunities of easily confused entities.

Model	WN18		FB15K	
	MR	Hit@10	MR	Hit@10
TransE	251	.892	125	.471
DC-TransE	265	.947	68	.853

Table 2: Entity Prediction Experiment Results compared DC-TransE with TransE.

Model	WN18RR		FB15K237		YAGO3-10	
	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10
DistMult	.397	.502	.224	.490	.413	.661
Complex	.425	.521	.257	.530	.505	.704
ConvE	.390	.508	.219	.476	.400	.658
RotatE	.426	.574	.238	.533	.405	.670
DC-RotatE	.425	.582	.246	.535	.419	.681

Table 3: Entity Prediction Experiment Results compared DC-RotatE with mainstream KRL

(3) Our method can effectively improve the modeling ability of RMP. Similarly, Table 4 reports the detailed RMP results of DC-TransE and TransE on FB15K. In all types of RMP, DC-TransE improves over TransE, especially in the challenging tasks of tail prediction for 1-n and head prediction for n-1.

Task	RMPs	TransE	DC-TransE
Predicting Head/Tail	1-to-1	.437/.437	.894/.879

(Hits@10)	1-to-N	.657/.197	.972/.671
	N-to-1	.182/.667	.567/.964
	N-to-N	.472/.500	.880/.910

Table 4: Comparing the RMP performance of DC-TransE and TransE on FB15k.

Similarly, in Table 5, we observe the same trend in the comparison of detailed RMP results of DC-RotatE and RotatE on FB15K237. This further validates that more distinctive training between entities with close semantic distances can effectively improve the prediction accuracy of 1-n and n-1, thus leading to better modeling ability of RMP.

Task	RMPs	RotatE	DC-RotatE
Predicting Head/Tail (MRR)	1-to-1	.498/.490	.516/.505
	1-to-N	.475/.071	.483/.089
	N-to-1	.088/.747	.107/.779
	N-to-N	.260/.367	.295/.386

Table 5: Comparing the RMP performance of RotatE and DC-RotatE on FB15k-237.

5 CONCLUSION

At present, knowledge representation learning techniques are improving in knowledge completion and semantic reasoning. However, they do not consider that the distribution of entities on the semantic plane of a particular relationship is unbalanced, and the semantic distance between entities has different effects on the training process. For example, when constructing a negative sample for a triple (Diego Armando Maradona, gender, male), it may produce a negative fact (Diego Armando Maradona, gender, female). But if we heavily choose low-value negative samples, such as (Diego Armando Maradona, gender, Cardiopathy) and (Diego Armando Maradona, gender, glycuerosis), it is not conducive to the improvement of training effect and prediction performance of knowledge representation model.

In order to solve the problem of low training efficiency caused by the imbalance of relationship and entity distribution in knowledge graph, this paper proposes a knowledge representation learning method based on the idea of training students to do multiple choice questions in daily life. The proposed approach involves conducting entity classification based on the "affinity and sparse" semantic distance of entities on a specific relation plane and adopting different training strategies. More training is given to entities that exhibit "affinity" (close semantic distance, dense distribution, and low differentiation), while less training is provided to those with "sparse" (far semantic distance, sparse distribution, and high differentiation). The loss function introduces different residual weights, adjusting the tensile rates of negative and positive samples with varying values by setting the weights. This approach aims to provide more training opportunities and lower tensile rates to the easily confused negative samples (high-value negative samples). Conversely, for easily recognized negative samples (low-value negative samples), fewer training opportunities are provided, and the tensile rate is improved. The use of different stretch weights serves as a balance to allocate different training opportunities to negative samples. Finally, the effectiveness of the proposed method is verified by comparing it with mainstream knowledge representation models.

Yi Zhang, <https://orcid.org/0000-0003-4409-9252>

Wanhua Cao, <https://orcid.org/0000-0002-7667-0153>

Juntao Liu, <https://orcid.org/0000-0003-1925-4484>

Yuanbin Wang, <https://orcid.org/0000-0002-4908-3869>
 Ziyun Rao, <https://orcid.org/0009-0003-6699-2276>

REFERENCES

- [1] Suchanek, F. M.; Kasneci, G.; Weikum, G.: Yago: a core of semantic knowledge, In Proceedings of the 16th international conference on World Wide Web, 2007, 697–706. <https://doi.org/10.1145/1242572.1242667>
- [2] Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor J.: Freebase: a collaboratively created graph database for structuring human knowledge, In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008, 1247–1250. <https://doi.org/10.1145/1376616.1376746>
- [3] Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives Z.: Dbpedia: A nucleus for a web of open data, In The semantic web, 2007, 722–735. https://doi.org/10.1007/978-3-540-76298-0_52
- [4] Vrandečić, D.; Krötzsch, M.: Wikidata: a free collaborative knowledgebase, Communications of the ACM, 57(10), 2014, 78–85. <https://doi.org/10.1145/2629489>
- [5] Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E. R.; Mitchell, T. M.: Toward an architecture for never-ending language learning, in Proceedings of AAAI, 2010, 1306–1313. <https://doi.org/10.1609/aaai.v24i1.7519>
- [6] Rotmensch, M.; Halpern, Y; Tlimat, A.; Horng, S.; Sontag, D.: Learning a Health Knowledge Graph from Electronic Medical Records, Sci Rep, 7(1), 2017, 5994. <https://doi.org/10.1038/s41598-017-05778-z>
- [7] Ji, S.; Pan, S.; Cambria, E.; Marttinen P.; Yu, P. S.: A Survey on Knowledge Graphs: Representation, Acquisition, and Applications, IEEE Transactions on Neural Networks and Learning Systems, 33(2), 2022, 494–514. <https://doi.org/10.1109/TNNLS.2021.3070843>
- [8] Bouarroudj, W.; Boufaïda, Z.; Bellatreche, L.: Named entity disambiguation in short texts over knowledge graphs, Knowl Inf Syst 64, 2022, 325–351. <https://doi.org/10.1007/s10115-021-01642-9>
- [9] Rao, Z.-Y.; Zhang, Y.; Liu J.-T., Cao Wan-Hua. Recommendation methods and systems using knowledge graph, Acta Automatica Sinica, 47(9), 2021, 2061–2077. <https://doi.org/10.16383/j.aas.c200128>
- [10] Sun, R.; Cao, X.; Zhao, Y.; Wan, J.; Zhou, K.; Zhang, F.; Wang, Z.; Zheng, K.: Multi-modal knowledge graphs for recommender systems, In International Conference on Information and Knowledge Management, 2020, 1405–1414. <https://doi.org/10.1145/3340531.3411947>
- [11] Dietz, L.; Xiong, C.; Dalton, J. et al.: Special issue on knowledge graphs and semantics in text analysis and retrieval, Inf Retrieval J., 22, 2019, 229–231. <https://doi.org/10.1007/s10791-019-09354-z>
- [12] Saxena, A.; Tripathi, A.; Talukdar, P.: Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings, In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, 4498–4507. <https://doi.org/10.18653/v1/2020.acl-main.412>
- [13] Cao, J.; Fang, J.; Meng, Z.; Liang, S.: Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces, ArXiv, 2022, <https://doi.org/10.48550/arXiv.2211.03536>
- [14] Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data, Neural Information Processing Systems, 2013, 2787–2795. <https://dl.acm.org/doi/10.5555/2999792.2999923>
- [15] Zhen, W.; Zhang, J.; Feng, J.; Zheng, C.: Knowledge Graph Embedding by Translating on Hyperplanes, In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, 1112–1119. <https://doi.org/10.1609/aaai.v28i1.8870>

- [16] Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion, In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, 2181–2187. <https://doi.org/10.1609/aaai.v29i1.9491>
- [17] Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, R.; Bouchard, G.: Complex embeddings for simple link prediction, In Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2016, 2071–2080. <https://dl.acm.org/doi/10.5555/3045390.3045609>
- [18] Yang, B.; Yih, W. T.; He, X.; Gao, J.; Deng, L.: Embedding entities and relations for learning and inference in knowledge bases, in ICLR, 2015, 1–13. <https://doi.org/10.48550/arXiv.1412.6575>
- [19] Sun, Z.; Deng, Z. H.; Nie, J. Y.; Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space, in ICLR, 2019, 1–18. <https://doi.org/10.48550/arXiv.1902.10197>
- [20] Ji, G.; He, S.; Xu, L.; Kang, L.; Zhao, J.: Knowledge Graph Embedding via Dynamic Mapping Matrix, in ACL-IJCNLP, 2015, 687–696. <https://doi.org/10.3115/v1/p15-1067>
- [21] Nickel, M.; Rosasco, L.; Poggio, T.: Holographic Embeddings of Knowledge Graphs, in AAAI, 2016, 1955–1961. <https://doi.org/10.1609/aaai.v30i1.10314>
- [22] Liu, H.; Wu, Y.; Yang, Y.: Analogical inference for multi-relational embeddings, in ICML, 2017, 2168–2178. <https://dl.acm.org/doi/10.5555/3305890.3305905>
- [23] Zhang, S.; Tay, Y.; Yao, L.; Liu, Q.: Quaternion knowledge graph embeddings, in NeurIPS, 2019, 2735–2745. <https://dl.acm.org/doi/10.5555/3454287.3454533>
- [24] He, S.; Kang, L.; Ji, G.; Zhao, J.: Learning to Represent Knowledge Graphs with Gaussian Embedding, in CIKM, 2015, 623–632. <https://dl.acm.org/doi/10.1145/2806416.2806502>
- [25] Han, X.; Huang, M.; Zhu, X.: TransG: A Generative Model for Knowledge Graph Embedding, in ACL, 2016, 2316–2325. <https://doi.org/10.18653/v1/P16-1219>
- [26] Han, X.; Huang, M.; Yu, H.; Zhu, X.: From one point to a manifold: orbit models for knowledge graph embedding, in IJCAI, 2016, 1315–1321. <https://dl.acm.org/doi/10.5555/3060621.3060804>
- [27] Ebisu, T.; Ichise, R.: TorusE: Knowledge Graph Embedding on a Lie Group, in AAAI, 2018, 1819–1826. <https://doi.org/10.1609/aaai.v32i1.11538>
- [28] Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y.: Learning structured embeddings of knowledge bases, In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence. 2011, 301–306. <https://doi.org/10.1609/aaai.v25i1.7917>
- [29] Xiao, H.; Huang, M.; Hao, Y.; Zhu, X.: Transa: an adaptive approach for knowledge graph embedding, in AAAI, 2015, 1–7. <https://doi.org/10.48550/arXiv.1509.05490>
- [30] Bordes, A.; Glorot, X.; Weston, J.; Bengio, Y.: A semantic matching energy function for learning with multi-relational data, Machine Learning, 94(2), 2014, 233–259. <https://doi.org/10.1007/s10994-013-5363-6>
- [31] Zhang, W.; Paudel, B.; Zhang, W.; Bernstein, A.; Chen, H.: Interaction embeddings for prediction and explanation in knowledge graphs, In ACM International Conference on Web Search and Data Mining, 2019, 96–104. <https://doi.org/10.1145/3289600.3291014>
- [32] Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S.: Convolutional 2d knowledge graph embeddings, in AAAI, 2018, 1811–1818. <https://doi.org/10.1609/aaai.v32i1.11573>
- [33] Cai, L.; Wang, W. Y.: KBGAN: Adversarial Learning for Knowledge Graph Embeddings, in NAACL, 2018, 1470–1480. <https://doi.org/10.18653/v1/n18-1133>
- [34] Miller, George, A.: Wordnet: a lexical database for English, Communications of the ACM, 38(11), 1995, 39–41. <https://dl.acm.org/doi/10.1145/219717.219748>
- [35] Toutanova, K.; Chen, D.: Observed Versus Latent Features for Knowledge Base and Text Inference, in ACL Workshop on CVSC, 2015, 57–66. <https://doi.org/10.18653/v1/w15-4007>