A Novel Negative Sampling Based on Frequency of Relational Association Entities for Knowledge Graph Embedding

Yi Zhang[1,2], Wanhua Cao[1,2,*], Juntao Liu[2] and Ziyun Rao[2]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China*
[2]*Wuhan Digital Engineering Research Institute, Wuhan, 430205, China*
E-mail: yzhang85@hrbeu.edu.cn
[*]*Corresponding Author*

Received 12 May 2021; Accepted 25 June 2021;
Publication 14 October 2021

Abstract

Knowledge graph embedding improves the performance of relation extraction and knowledge reasoning by encoding entities and relationships in low-dimensional semantic space. During training, negative samples are usually constructed by replacing the head/tail entity. And the different replacing relationships lead to different accuracy of the prediction results. This paper develops a negative triplets construction framework according to the frequency of relational association entities. The proposed construction framework can fully consider the quantitative of relations and entities in the dataset to assign the proportion of relation and entity replacement and the frequency of the entities associated with each relationship to set reasonable proportions for different relations. To verify the validity of the proposed construction framework, it is integrated into the state-of-the-art knowledge graph embedding models, such as TransE, TransH, DistMult, ComplEx, and Analogy. And both the evaluation criteria of relation prediction and entity prediction are

Journal of Web Engineering, Vol. 20_6, 1867–1884.
doi: 10.13052/jwe1540-9589.2068

used to evaluate the performance of link prediction more comprehensively. The experimental results on two commonly used datasets, WN18 and FB15K, show that the proposed method improves entity link and triplet classification accuracy, especially the accuracy of relational link prediction.

Keywords: Knowledge embeddings, negative sampling, link prediction, knowledge graph.

1 Introduction

In recent years, knowledge graphs have been used as an important resource in many fields [1], such as interactive retrieval [2], intelligence analysis [3] and intelligent question answering [4]. Some famous knowledge graphs, such as FreeBase [5] and YAGO [6], store vast amounts of structured data usually in the form of triplets (***head entity***, ***relation***, ***tail entity***) (abridged as (***h***, ***r***, ***t***)), each of which indicates that there is a ***relation*** between the ***head*** and the ***tail*** in the real world. For example, the triple (***Beijing***, ***the capital of***, ***China***) indicates that there is a relationship between the head entity "***Beijing***" and the tail entity "***China***" as "***the capital of***". Although current knowledge graphs contain millions of entities and billions of relational facts, there are still huge amounts of unobserved relational facts. As an efficient way to automatically predict the unknown relational facts, knowledge representation learning completes knowledge graphs. It embeds the semantic information of the entities and relations in knowledge graphs into the dense low-dimensional real-valued vectors, which can fully reveal the semantic relations between the entities and the relations to effectively alleviate the data sparseness caused by the long tail distribution of the knowledge graphs. The semantic similarity between entities is fast calculated as the distance between two vectors in low-dimensional vector space to improve the computational efficiency of knowledge representation learning in knowledge graphs. Because of the advantage, many methods of knowledge representation learning are proposed. These methods are usually divided into two categories: Tensor Factorization Based Methods and Mapping Based Methods. The former contains DistMult [7], ComplEx [8], ANALOGY [9], etc., and the latter contains, Semantic Matching Energy (SME) [10], Semantically Smooth Embedding (SSE) [11], and translation-based methods [12–19].

Translation-based methods, such as TransE [12], TransH [13] and TransR [14], are the most used knowledge graphs embedding methods because of their simplicity and high efficiency. A relation is regarded as a

translation from the head entity to the tail entity in these methods. In this kind of method, usually, a score function with parameters is defined, and the margin-based training objective is used to separate a positive triplet and its corresponding negative triplet. But there is a difficulty in training the translation-based models. There are only positive training triplets in knowledge graphs. Usually, to solve this problem, a negative triplet is constructed by replacing the head or tail entity of a positive triplet randomly. Because of the simplicity and easy implementation, it's widely used by some translation-based knowledge representation models, such as TransE, TransH, TransR, and TransD [15], and other kinds of models, such as SE, SME, NTN. However, a practical knowledge graph is often far from complete, and this simple randomly replacing method may introduce many false-negative triplets in the training process. Here, the false-negative triplet is the positive triplets that are mistakenly treated in training as a negative example. Because it is not possible to traverse all the positive triplets when constructing negative examples, false-negative triplets can't be avoided, and they can only be reduced by reasonable sampling. TransH improves the negative triplets constructing method. In TransH, according to the category of relations, such as 1-to-1, 1-to-N, N-to-1 and N-to-N, the head or tail entities are replaced in different probabilities when constructing negative triplets. Thus, the head entity in one-to-many relations will get more chance to be replaced, and the tail entity in many-to-one relations will get more chance to be replaced. This improvement, widely used in TransR, TransD and TranSparse [18], can effectively reduce the probability of generating false-negative triplets, and it also improves the accuracy of the entity link prediction which aims to complete a triplet (h, r, t) with h or t missing. To achieve better results in the relation link prediction which aims to complete a triplet (h, r, t) with r missing, some presentation models, such as TransA [16], TransG [19], replace the relation of a triplet to construct the negative example for training, and improve the accuracy of the relation link prediction. However, because of the imbalance of randomly sampling and the unsuitable proportion of replacing probabilities between the relation and entity in a triplet, it may introduce many false-negative triplets into training and lead to some errors for link predictions.

In this paper, a novel negative sample generating framework in the training of knowledge representation model is proposed. The remainder of this paper is organized as follows. Section 2 elaborates on the negative triplets construction framework proposed in this paper. And in Section 3, the performance of the knowledge representation model of the proposed negative triplets construction framework is tested, and compared with the

original construction framework. Finally, in Section 4 the conclusions and future research direction are given.

2 Negative Samples Construction Framework Based on Frequency of Relational Association Entities

2.1 Negative Samples Generating Strategy

In the process of negative example construction, Excessive false-negative samples reduce the accuracy of the representation model, and Too many worthless negative triplets make the model convergence slow. To alleviate the two problems above, a novel negative sampling framework is proposed. It sets different probabilities to replace the three elements in the triplet: the head entity, the relationship, and the tail entity. As replacing two or more elements in a triplet at the same time will greatly increase the probability of occurrence of false-negative triplets, only one element of the triple is replaced at a time in our work. In addition, the sum of the replacement probabilities of the relationship, the head entity and the tail entity equal to 1.

First, a formal description of the set of negative triplets constructed by this framework is as follows:

$$S'_{(h,r,t)\in S} = \{(h',r,t)|h' \in E\} \cup \{(h,r',t)|r' \in R\} \cup \{(h,r,t')|t' \in E\} \tag{1}$$

where $S'_{(h,r,t)}$ (Shortly referred to as S') is a set of corrupted triplets constructed by replacing the head entity h in the triplet (h,r,t) with h' or replacing the tail entity t with t' or replacing the relation r with r', and the replaced triplets (h',r,t), (h,r,t') and (h,r',t) are restricted not in the original $S_{(h,r,t)}$.

Next, we study the relationship replacement probability assigned to the triplet. There are differences between each relationship. For example, some relationships is associated with few entities, such as the relationship numbered 139 in the FB15k dataset: */metropolitan_transit/transit_service_type/transit_lines*, only two entities related with it: Entity number 529: */m/0195fx*, entity number 530: */m/0m_sb*; the relationship numbered 1028: */user/tsegaran/computer/algorithm/family*, and relationship number 1225: */user/tsegaran/computer/algorithm_family/algorithm*, there is only one entity related with them, 6917:*/m/0382k*.

Some relationships have many related entities, such as relationship 30: */people/person/profession*, there are 4296 entities related with it.

It is not appropriate to use the same replacement probability for the above two types of relationships. Therefore, for different relationships, different relationship-entity replacement probability assignment values need to be set. For triplets containing relationships with a small number of related entities, we give them more opportunities to replace the entities; for those triplets containing relationships with a larger number of related entities, we give them more opportunities to replace relationships.

Specifically, we traverse the training set and get three statistics for all triples containing specific relation r:

- the number of entities related with r: $C_{Ent}(r)$;
- the average number of tail entities per head entity, denoted as $tph(r)$;
- the average number of head entities per tail entity, denoted as $hpt(r)$.

Considering the other two features of the data set: the total number of relationships $|R|$ and the total number of entities $|E|$, we set the probability of the replacement of the triplets containing the relationship r:

$$P_r = \frac{C_{Ent}(r)}{1 + C_{Ent}(r)} \times \frac{|R|}{|E| + |R|} \tag{2}$$

The greater entities related with the relationship r, the closer the relation replacement probability of the triplet is to $\frac{|R|}{|E|+|R|}$ when constructing a negative example; The less entities related with the relationship r, the closer the relation replacement probability of the triplet is to $\frac{1}{2} \cdot \frac{|R|}{|E|+|R|}$. Because the triplet containing the relationship r has appeared in the training set, at least one entity is associated with it, that is, $C_{Ent}(r) \geq 1$. Under special circumstances, there is $C_{Ent}(r) = 1$, that is, the head and tail entities related with the relationship r are same. For example, the numbered 1225 relationship in the FB15k: */user/tsegaran/computer/algorithm_family/algorithm*, the number of its related entities is 1.

Finally, the probability allocation method for replacing head and tail entities of TransH is extended as

$$\frac{tph(r)}{tph(r) + hpt(r)} \tag{3}$$

Which is defined as a parameter for sampling: for a true triplet (h, r, t) with relation r, and the replacement probability assigned to the head

entity is

$$P_h = (1 - P_r)\frac{tph(r)}{tph(r) + hpt(r)} \tag{4}$$

the replacement probability assigned to the tail entity is

$$P_t = (1 - P_r)\frac{hpt(r)}{tph(r) + hpt(r)} \tag{5}$$

2.2 The Probability of Generating False-negative Samples in the Proposed Framework

For an element in the set of constructed triplets, $\Delta' \in S'$, if Δ' is true, that is, it is a false-negative triplet, and this situation is recorded as event A. if Δ' is a valid negative triplet, and this situation is recorded as event $\overline{A}$.

The element of events set $B = \{B_r, B_h, B_t\}$ denotes the event which method is selected to generate the negative triplets, relation replacement, head entity replacement, or tail entity replacement. B_r, B_h, B_t is a division of the sample space B, expressed as Equation (6).

$$\begin{cases} P(B_r) + P(B_h) + P(B_t) = 1 \\ B_r \cap B_h = \phi \\ B_r \cap B_t = \phi \\ B_h \cap B_t = \phi \end{cases} \tag{6}$$

According to Total Probability Theorem , the probability of generating false-negative triplets using head and tail entity replacement and relationship replacement methods can be calculated by Equation (7).

$$P(A) = P(A|B_r)P(B_r) + P(A|B_h)P(B_h) + P(A|B_t)P(B_t) \tag{7}$$

where $P(A|B_r)$ represents the probability that the triplet (h, r', t) obtained through relationship replacement is true, that is, the probability of obtaining an invalid negative triplet through relationship replacement; $P(A|B_h)$ represents the probability of generating false-negative triplets by head entity replacing; $P(A|B_t)$ represents the probability of generating false-negative triplets by tail entity replacing; $P(B_r)$ represents the probability of using relationship replacement when constructing corrupted triplets; $P(B_h)$ represents the probability of using head entity replacement when constructing corrupted triplets; $P(B_t)$ represents the probability of using tail entity replacement when constructing corrupted triplets.

There are $|R|-1$ possible values for the triplet (h, r', t) after the relationship replacement of the triplet (h, r, t), because $r' \in R, r \neq r'$. The number of false-negative triplets is the number of constructed triplets appeared in the dataset. The number of occurrences of the relationship for the fixed (h, t) in the dataset can be counted, and recorded it as $rp(h, t)$. Therefore, the probability of invalid negative triplets $P(A|B_r)$ produced by the relationship replacement can be expressed by Equation (8).

$$P(A|B_r) = \frac{rp(h,t) - 1}{|R| - 1} \tag{8}$$

where both the numerator and denominator minus 1 is due to the removal of the original triple (h, r, t). It should be noted that the triplets in the dataset are not repeated, and we only count the pairs (h, t) that appear in the triplets in the dataset. So, $1 \leq rp(h, t) \leq |R|$.

There are $|E| - 1$ possible values for the triplet (h', r, t) after the head entity replacement of the triplet (h, r, t), because $h' \in E, h \neq h'$. The number of invalid negative triplets is the number of constructed triplets appeared in the dataset. The number of head entities that appear in each tail entity for the relation r in the dataset can be counted and denoted as $hpt(r)$. Therefore, the probability of false-negative triplets being replaced by the head entity $P(A|B_h)$ can be expressed as Equation (9). Same as above that the numerator and denominator minus 1 because the head entity$h$in the original triple is going to be removed.

$$P(A|B_h) = \frac{hpt(r) - 1}{|E| - 1} \tag{9}$$

There are $|E| - 1$ possible values for the triplet (h, r, t') after the head entity replacement of the triplet (h, r, t). This is because $t' \in E, t \neq t'$. The number of invalid negative triplets is the number of constructed triplets appeared in the dataset. The number of tail entities that appear in each tail entity for the relation r in the dataset can be counted and denoted as $tph(r)$. Therefore, the probability of false-negative triplets being replaced by the tail entity $P(A|B_t)$ can be expressed by Equation (10). Same as above that the numerator and denominator minus 1 because the tail entity t in the original triple is going to be removed.

$$P(A|B_t) = \frac{tph(r) - 1}{|E| - 1} \tag{10}$$

In consequence, the probability of generating false-negative triplets $P(A)$ using head and tail entity replacement and relationship replacement methods can be calculated by Equation (11).

$$P(A) = \frac{rp(h,t)-1}{|R|-1}P(B_r) + \frac{hpt(r)-1}{|E|-1}P(B_h) + \frac{tph(r)-1}{|E|-1}P(B_t) \tag{11}$$

where the sum of $P(B_r)$, $P(B_h)$ and $P(B_t)$ equal to 1. And in the proposed framework, $P(B_r)$, $P(B_h)$ and $P(B_t)$ is defined as follow:

$$\begin{cases} P(B_r) = P_r = \dfrac{C_{Ent}(r)}{1+C_{Ent}(r)} \times \dfrac{|R|}{|E|+|R|} \\ P(B_h) = P_h = (1-P_r)\dfrac{tph(r)}{tph(r)+hpt(r)} \\ P(B_t) = P_t = (1-P_r)\dfrac{hpt(r)}{tph(r)+hpt(r)} \end{cases} \tag{12}$$

Finally, the probability of generating false-negative triplets $P(A)$ can be obtained:

$$\begin{aligned} P(A) &= \frac{rp(h,t)-1}{|R|-1}P(B_r) + \frac{hpt(r)-1}{|E|-1}P(B_h) + \frac{tph(r)-1}{|E|-1}P(B_t) \\ &= \frac{rp(h,t)-1}{|R|-1}P_r + \frac{hpt(r)-1}{|E|-1}(1-P_r)\frac{tph(r)}{tph(r)+hpt(r)} \\ &\quad + \frac{tph(r)-1}{|E|-1}(1-P_r)\frac{hpt(r)}{tph(r)+hpt(r)} \\ &= \frac{rp(h,t)-1}{|R|-1}P_r + (1-P_r)\frac{2tph(r)\cdot htp(r) - (tph(r)+htp(r))}{(|E|-1)(tph(r)+hpt(r))} \\ &= \frac{rp(h,t)-1}{|R|-1}P_r + (1-P_r) \\ &\quad \times \left(\frac{2tph(r)\cdot htp(r)}{(|E|-1)(tph(r)+hpt(r))} - \frac{1}{|E|-1}\right) \end{aligned} \tag{13}$$

3 Experimental Results and Discussion

In this section, the proposed negative samples generating framework is verified on datasets FB15K and WN18 by entity prediction and relation prediction tasks. The impact of the probability of relation replacement is also investigated. First, the datasets FB15K and WN18 are introduced. Then, in the experimental setup, our evaluation protocol and implementation are described. In order to analyse the impact of replacing probability on the entity and relation prediction, we select the relation replacement probabilities from a list of real values between 0 and 1.0. The experimental results show the effectiveness of our improvement in the negative samples generating framework.

3.1 Datasets

To make objective experimental comparisons with more representation models, two widely chosen datasets FB15K and WN18 [20] are used to evaluate our framework.

Among them, FB15K is a data set extracted from Freebase, which contains 14,951 entities, 1,345 relationships and 592,213 triplets; WN18 is a data set extracted from WordNet, which contains 40,943 entities, 18 relationships and 151,442 triplets. The partitioning of these two datasets and their training set, validation set and test set have been published [20]. Please refer to Table 1 for details.

3.2 Results

For link prediction, relational link prediction, and triplet classification testing, the same model uses the same set of hypermeters on a data set. The parameter settings are shown in Table 2. Among them, "+" means that our negative example construction framework is applied to the model on the left.

The Entity prediction results is presented in Table 3. The Best Results for Each Case are bolded,and "−" denote as no result in the original paper. The

Table 1 Datasets used in the experiment

Dataset	#Rel	#Ent	#Train	#Valid	#Test
FB15K	1,345	14,951	483,142	50,000	59,071
WN18	18	40,943	141,442	5,000	5,000

Table 2 Hyperparameters used in the model in the experiment

Model	WN18	FB15K
TransE	d=100, $\gamma = 1$, $\alpha = 0.001$	d = 100, $\gamma = 4$, $\alpha = 0.001$
TransE+SparseNSG	d = 100, $\gamma = 1$, $\alpha = 0.001$	d = 100, $\gamma = 4$, $\alpha = 0.001$
TransH	d = 100, $\gamma = 1$, $\alpha = 0.001$	d = 100, $\gamma = 4$, $\alpha = 0.001$
TransH+SparseNSG	d = 100, $\gamma = 1$, $\alpha = 0.001$	d = 100, $\gamma = 4$, $\alpha = 0.001$
DISTMULT	d = 200, $\alpha = 0.1$	d = 200, $\alpha = 0.1$
DISTMULT+SparseNSG	d = 200, $\alpha = 0.1$	d = 200, $\alpha = 0.1$
COMPLEX	d = 200, $\alpha = 0.1$	d = 200, $\alpha = 0.1$
COMPLEX+SparseNSG	d = 200, $\alpha = 0.1$	d = 200, $\alpha = 0.1$
ANALOGY	d = 200, $\alpha = 0.1$	d = 200, $\alpha = 0.1$
ANALOGY+SparseNSG	d = 200, $\alpha = 0.1$	d = 200, $\alpha = 0.1$

relational prediction results are listed in Table 4, and Table 5 shows the triple classification results. The Best Results for Each Case are bolded, too.

3.3 Discussion

According to the comparison results of the entity predictions listed in Table 3, it can be seen that the proposed negative triplets construction framework can improve the entity prediction performance on the majority of knowledge presentation models. The entire training round is the same, and the new negative example construction method divides the probability of replacing the head and tail entities into a part for relationship replacement for training. The number of training rounds that are originally given to the replacement of head and tail entities has been reduced, which may lead to a decrease in the performance of entity prediction. However, the experimental data of Table 3 shows that the performance of the entity has not decreased, and the performance of entity prediction has improved. The reason is that the training of constructing negative examples of relationship replacement is not completely independent of the entity's predictive ability. The increase in the constructive dimension of negative examples can affect the entity's predictive ability, which is the result of adding a reasonable probability of relationship replacement.

In addition, in terms of the number of entities, WN18 has 40,943 and FB15K has 14,951. Entity prediction is to select from so many entities the entities that can be missing from the triple. Therefore, in Table 3, hit@10 is

Table 3 Comparison of entity prediction experiment results

	WN18						FB15K					
	MRR		MeanRank		Hit@10(%)		MRR		MeanRank		Hit@10(%)	
Datasets Model	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	0.335	0.454	263	251	75.4	89.2	0.221	0.38	243	125	34.9	47.1
TransE+SparseNSG	**0.434**	**0.596**	**215**	**199.5**	72.9	82.6	**0.277**	**0.509**	**146.5**	**47.9**	**52.3**	**74.1**
TransH	–	–	400.8	388	73.0	82.3	–	–	212	87	45.7	64.4
TransH+SparseNSG	0.433	0.591	**203**	**188.2**	**75.2**	**85.8**	0.276	0.508	**146.1**	**47.5**	**52.5**	**74.1**
DISTMULT	0.532	0.882	–	–	–	93.6	0.242	0.654	–	–	–	82.4
DISTMULT+SparseNSG	**0.535**	0.830	300.7	285	79.4	**94.0**	**0.276**	0.65	164	41.4	52.5	**82.7**
COMPLEX	0.587	0.941	**–**	–	–	94.7	0.242	0.692	–	–	–	84.0
COMPLEX+SparseNSG	0.586	**0.942**	343	327	80.1	94.5	**0.26**	0.686	175	42.7	50.3	82.7
ANALOGY	0.657	0.942	**–**	–	–	94.7	0.253	0.725	–	–	–	85.4
ANALOGY+SparseNSG	0.571	**0.940**	352	336	79.6	94.2	**0.261**	0.693	175.4	42.9	50.2	82.7

Table 4 Comparison of relationship prediction experiment results

	WN18				FB15K			
	MeanRank		Hit@3(%)		MeanRank		Hit@3(%)	
Datasets Model	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	3.27	3.27	73.6	73.64	208	207.6	58.2	58.8
TransE+SparseNSG	**3.14**	**3.14**	**74.8**	**74.8**	**3.28**	**2.93**	**89.4**	**93.7**
TransH	3.34	3.34	70.1	70.1	212.8	212.4	44.9	48.2
TransH+SparseNSG	**2.92**	**2.92**	**73.6**	**73.6**	**3.22**	**2.87**	**89.8**	**94**
DISTMULT	3.18	3.02	72.6	72.6	55.9	53.6	89.9	95
DISTMULT+SparseNSG	**2.24**	**2.96**	**73.5**	**74.9**	**34.6**	**32.6**	**92.6**	**97**
COMPLEX	3.26	3.16	74.6	75.7	52.6	51.3	89.6	93.5
COMPLEX+SparseNSG	**3.15**	**2.97**	**75.1**	**76.2**	**33.71**	**33.24**	**92.6**	**94.6**
ANALOGY	3.69	3.57	75.8	76.9	55.7	54.3	86.8	89.4
ANALOGY+SparseNSG	**3.58**	**3.49**	**76.5**	**78.3**	**31.7**	**30.5**	**92.6**	**95.4**

suitable for comparing the prediction performance of different model entity links.

The amount of relationship coefficients between the two data sets is much smaller than the number of entities, FB15K only has 1,345, and WN18 even has only 18. Hit@10 is no longer suitable for evaluating the predictive performance of the model. Therefore, in Table 4, hit@3 is more suitable for evaluating the predictive performance of relational links.

The comparison results in Table 4 show that the relationship prediction performance has improved on the WN18 dataset, but not as obvious as FB15K. This is because the number of relationships in FB15K is 1,345, while the number of relationships in WN18 is only 18. The FB15K dataset has a wide variety of relationships and is more complicated. Therefore, the relationship prediction on FB15K is much more difficult than WN18. Experiments on two datasets show that in relational prediction, the performance of relation prediction has been significantly improved after applying the proposed negative triplets construction framework to these knowledge representation models. This shows that our negative triplets construction framework is more comprehensive than the previous negative triplets construction in the training of the model. It fully demonstrates the influence of the relationship vector on the performance of training in the knowledge representation model.

In Table 5, the triple classification experiment results demonstrate that the proposed negative triplets construction framework can improve the

Table 5 Comparison of the results of the triple classification experiment

Model	FB15K
TransE	79.6
TransE+SparseNSG	**83.0**
TransH	80.2
TransH+SparseNSG	**83.1**
DISTMULT	87.0
DISTMULT+SparseNSG	**87.7**
COMPLEX	87.2
COMPLEX+SparseNSG	**88.6**
ANALOGY	87.8
ANALOGY+SparseNSG	**89.5**

performance of the knowledge representation model on the triple classification task. In order to further discuss the effectiveness of the proposed negative triplets construction framework on the triple classification task.

4 Conclusion

In this paper, a new negative triplets construction framwork based on relational association entity sparseness is presented and applied to knowledge representation learning models TransE, TransH, ComplEx, DistMult and ANALOGY. The framework optimized the training process of knowledge representation learning by introducing the statistical features of the relationship between the datasets and the association between entities into the distribution probability of the relationship replacement in the negative case construction, achieved an improvement in the performance of entity and relationship link prediction, especially the performance improvement of relationship prediction is more significant. Moreover, we study the impact of the probability ratio of relationship replacement and entity replacement in the negative triplets construction on entity prediction and relationship prediction performance. The experimental results show that the ratio of relationship replacement and entity replacement probability which is designed based on dataset relationships and statistical characteristics in our negative triplets construction framework can make the multi-class typical knowledge representation learning model achieve a balance between relationship prediction and entity prediction performance. However, the influence of the ratio of

positive and negative samples in training is not considered in this new framework.

The above knowledge representation learning models perform well when predicting the tasks of entities and relationships that have occurred in the training set. And the performance of the forecasting tasks for new entities and relationships is not ideal. Many knowledge representation models that can handle new entities and relationships will be proposed. In the future, how to quickly and accurately predict the open representation model of newly entered entities and relationships will be an important development direction; it is also worthwhile to study whether the open representation model can continue to reuse the negative triplets construction framework proposed in this paper; the replacement probability assignment of the relationship with the relationship, the actual scenario match the relationship prediction and the match between actual scenario and the relationship prediction and the entity prediction requirements are also the directions to be studied next.

References

[1] Y. Dai, S. Wang, N. Xiong, W. Guo, 'A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks', Electronics, 2020.

[2] J. Liu, 'Deconstructing search tasks in interactive information retrieval: A systematic review of task dimensions and predictors', Information Processing & Management, 2021, 58(3):102522.

[3] B. Shao, X. Li, G. Bian, 'A Survey of Research Hotspots and Frontier Trends of Recommendation Systems from the Perspective of Knowledge Graph', Expert Systems with Applications, 2020(165):113764.

[4] X. Li, H. Zang, X. Yu, et al. 'On improving knowledge graph facilitated simple question answering system', Neural Computing and Applications, 2021(2).

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, 'Freebase: A collaboratively created graph database for structuring human knowledge', in Proc. of ACM SIGMOD Int. Conf. on Manage. Data, 2008, pp. 1247–1250.

[6] F.M. Suchanek, G. Kasneci, and G. Weikum, 'YAGO: A core of semantic knowledge', in Proc. 16th Int. Conf. on World Wide Web, 2007, pp. 697–706.

[7] B. Yang, WT. Yih, X. He, J. Gao, and L. Deng, 'Embedding entities and relations for learning and inference in knowledge bases', in Proc. Int. Conf. Learn. Represent., 2015.
[8] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, 'Complex embeddings for simple link prediction', in Proc. 33rd Int. Conf. Mach. Learn., 2016, pp. 2071–2080.
[9] H. Liu, Y. Wu, and Y. Yang, 'Analogical inference for multirelational embeddings,' in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 2168–2178.
[10] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, 'A semantic matching energy function for learning with multi-relational data', Mach. Learn., vol. 94, no. 2, pp. 233–259, 2014.
[11] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, 'Semantically smooth knowledge graph embedding', in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Language Process., 2015, pp. 84–94.
[12] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, 'Translating embeddings for modeling multi-relational data', in Adv. Neural Inf. Process. Syst., 2013, pp. 2787–2795.
[13] Z. Wang, J. Zhang, J. Feng, and Z. Chen, 'Knowledge graph embedding by translating on hyperplanes', in Proc. 28th AAAI Conf. Artif. Intell., 2014, pp. 1112–1119.
[14] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in Proc. 29th AAAI Conf. Artif. Intell., 2015, pp. 2181–2187.
[15] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, 'Knowledge graph embedding via dynamic mapping matrix', in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Language Process., 2015, pp. 687–696.
[16] H. Xiao, M. Huang, Y. Hao, and X. Zhu, 'TransA: An adaptive approach for knowledge graph embedding', in arXiv:1509.05490, 2015.
[17] S. He, K. Liu, G. Ji, and J. Zhao, 'Learning to represent knowledge graphs with Gaussian embedding', in Proc. 24th ACM Int. Conf. Inf. Knowl. Manage., 2015, pp. 623–632.
[18] G. Ji, K. Liu, S. He, and J. Zhao, 'Knowledge graph completion with adaptive sparse transfer matrix', in Proc. 30th AAAI Conf. Artif. Intell., 2016, pp. 985–991.

[19] H. Xiao, M. Huang, Y. Hao, et al. 'TransG : A Generative Mixture Model for Knowledge Graph Embedding', Computer Science, 2015.
[20] A. Bordes, X. Glorot, J. Weston, 'Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing', International Conference on Artificial Intelligence & Statistics, 2012.

Biographies

Yi Zhang is a Ph.D. student at the College of Computer Science and Technology, Harbin Engineering University (HEU) from Autumn 2015. He received his Bachelor's degree in School of Computer Science & Technology from Huazhong University of Science & Technology (HUST) in 2007, Master's degree in School of software engineering from University of Science and Technology of China (USTC) in 2011. He is now a senior engineer in Wuhan Digital Engineering Research Institute. He is currently completing a doctorate in Computer Science at the Harbin Engineering University. His research interest covers knowledge computing, knowledge graph, and database technology.

Wanhua Cao is a PhD supervisor at the College of Computer Science and Technology, Harbin Engineering University (HEU). He received his Bachelor's degree in Huazhong University of Science & Technology (HUST) and Master's degree in China Ship Research and Development Academy. He is a deputy director, researcher at Wuhan Digital Engineering Institute. His main research interest is decision support.

Juntao Liu received the BS and MS degrees both in Computer Science from Ordnance Engineering College, Shijiazhuang, China, in 2002 and 2005, respectively, and PhD degree in Communication and Information Systems from Huazhong University of Science & Technology (HUST), Wuhan, China, in 2014. He is now a senior engineer in Wuhan Digital Engineering Research Institute. His research interests include data mining, machine learning and computer vision.

Ziyun Rao is a Master student at Wuhan Digital Engineering Institute. She received her Bachelor's degree from Huazhong University of Science & Technology (HUST). Her research interest covers knowledge graph and recommendation system.