GCL-KGE: Graph Contrastive Learning for Knowledge Graph Embedding

Qimeng Guo, Huajuan Duan, Chuanhao Dong, Peiyu Liu(✉), and Liancheng Xu(✉)

Shandong Normal University, Jinan 250358, China
{lcxu,liupy}@sdnu.edu.cn

Abstract. Knowledge graph embedding models characterize entities and relations in structured knowledge graphs as vectors, which is essential for many downstream tasks. Some studies show that knowledge graph embedding models based on graph neural networks can exploit higher-order neighborhood information and generate meaningful representations. However, most models suffer from interference from distant neighborhood noise information. To address the challenge, we propose a graph contrastive learning knowledge graph embedding (GCL-KGE)model to enhance the representation of entities. Specifically, we use the graph attention network to aggregate multi-order neighbor information optimizing the pretrained entity representation. To avoid the inclusion of redundant information in the graph attention network, we combine contrastive learning to provide auxiliary supervised signals. A new method of constructing positive instances in contrastive learning makes the entity representation in the hidden layer produce a marked effect in this paper. We use a triple scoring function to evaluate representation on link prediction. The experimental results on four datasets show that our model can alleviate the interactive noise and achieve better results than baseline models.

Keywords: Knowledge graph · Contrastive learning · Graph attention networks

1 Introduction

The knowledge graph(KG) stores facts in the real world as graph structures, e.g., in the form of a triple: (*The Hours, starred_actors, Meryl Streep*). The facts in the knowledge graph are always incomplete and manual completion is time-consuming and laborious. One way to complete the knowledge graph is knowledge graph embedding(KGE), which is the process of embedding entities and relations of the knowledge graph into a continuous vector space while preserving the structural and semantic information.

Knowledge graph embedding models apply a scoring function to measure the confidence of triples. Earlier knowledge graph embedding models are traditionally divided into distance-based models and tensor decomposition-based models [1]. They have high computational efficiency or a strong ability to

M. Tanveer et al. (Eds.): ICONIP 2022, CCIS 1791, pp. 177–188, 2023.
https://doi.org/10.1007/978-981-99-1639-9_15

express the model. With the widespread application of neural networks, some researchers apply graph attention networks(GAT) [2] to enrich entity representations because of the ability to exploit higher-order neighbor information. More recently, a number of studies [3,4] demonstrate that contrastive learning has the superiority to train effective graph representation learning when given unlabelled graph data. Some studies attempt to apply contrastive learning to knowledge graph embedding model to mine semantic similarities between triples.

Inspired by the success of the above studies, we explore the technical application of graph contrastive learning to knowledge graph embedding. We identify two potential challenges in current knowledge graph embedding models. They are: (i) Most models usually deal with each entity independently and ignore the structural relations of neighborhood triples in the knowledge graph. Therefore they only marginally model the graph structure of the knowledge graph. (ii) The representation of entities is susceptible to interaction noise due to the extra information of the graph attention network extending to distant nodes with little relevance. Besides, how to avoid changing the triples semantics in the process of constructing positive instances is a challenging question for the success of contrastive learning in the knowledge graph embedding model.

In this work, we propose a graph contrastive learning knowledge graph embedding model(GCL-KGE) to address these challenges. An encoder-decoder framework combined with contrastive learning is used in our model which obtains the structure information of the knowledge graph while utilizing the interactive noise to optimize the representation. Specifically, we first use the GAT which gives attention with different levels to neighbors to associate entities with neighbors optimizing the pre-trained entity embedding. Then we use a scoring function of a convolutional neural network-based knowledge graph embedding model for link prediction to evaluate the level of embedding. To decrease the influence of interactive noise, we perform contrastive learning based on the GAT without data augmentation which will change the semantics of the triple. The core idea is taking GAT's hidden representations as positive instances which are semantically similar to the final entity representation. The knowledge graph embedding is learned by maximizing the consistency between different augmented views of the same data in the hidden space. Experimental studies on four datasets demonstrate the effectiveness of GCL-KGE, which significantly improves the accuracy.

In summary, our contributions are as follows:

1. We propose a graph contrastive learning knowledge graph embedding(GCL-KGE) model to improve accuracy and robustness of existing knowledge graph embedding models.
2. Our proposed contrastive learning architecture provides auxiliary supervision signals for knowledge graph embedding and we perform a theoretical derivation for the direction of entity representation in contrastive learning.
3. Experimental results show the effectiveness of our model on the knowledge graph link prediction.

2 Related Works

Knowledge graph embedding has become a popular research topic attracting a wide range of researchers. These methods both determine the reality of a triple by constructing a scoring function.

The traditional knowledge graph embedding methods are mainly divided into two types: the distance-based models and the tensor decomposition-based models. The distance-based model focuses on calculating the distance between entities to set the scoring function. TransE [6] is the most widely used of these and regards the relation vector as the translation between the head entity and the tail entity. Based on it, researchers propose more variants in complex relations, such as TransR [22], TransD [23]. The tensor decomposition-based models map head entities to tail entities by multiplying the relationship matrices. RESCAL [7] uses vectors to represent the latent semantics of the entities and matrices to represent relations to model the semantics between potential factors. To better model asymmetric matrices, ComplEx [14] extends the model to the complex space. However, these models separately optimize each triple with the scoring function, overlooking the relations between the triples.

Recent studies use neural networks to learn representations of knowledge graphs. ConvKB [5] uses the convolutional neural networks(CNN) to extract triple features for link prediction. R-GCN [8] applies graph convolution networks(GCN) to link prediction and assigns the same weight to the neighboring entities of each entity. To reflect the different importance of different relations for entities, SCAN [9] sets the weight of aggregated neighbor information related to the class of relations. Referring to the idea of the GAT, work [1] proposes aggregating the overall neighbor triple information to train the representation. Nevertheless, as the graph attention network hierarchy deepens, information from more distant entities is aggregated into the entity representation, which leads to the introduction of more noisy information.

Contrastive learning is treated as an instrumental part of self-supervised learning and it has ability to learn a good representation based on the data's characteristics. The goal of contrastive learning is to pull the semantically close pairs together and push apart the negative pairs. Some models often use data augmentation to construct positive and negative instances, such as image flip, rotation, and cutout in computer vision [10,11]. In natural language processing, some studies use sentence crop, span deletion and reordering [12,13]. But the triples in the knowledge graph are different from the sentences in other tasks. If we add random noise to the embedding space, the semantic of the original triple will be changed and the incompleteness of the knowledge graph will be deepened.

To address the above issues, we propose the GCL-KGE to learn the knowledge graph embedding. We apply the graph attention network in GCL-KGE to aggregate the neighbor triple information to cope with separate training. And we propose a new way to construct positive instances to solve noise interference without semantic deficits.

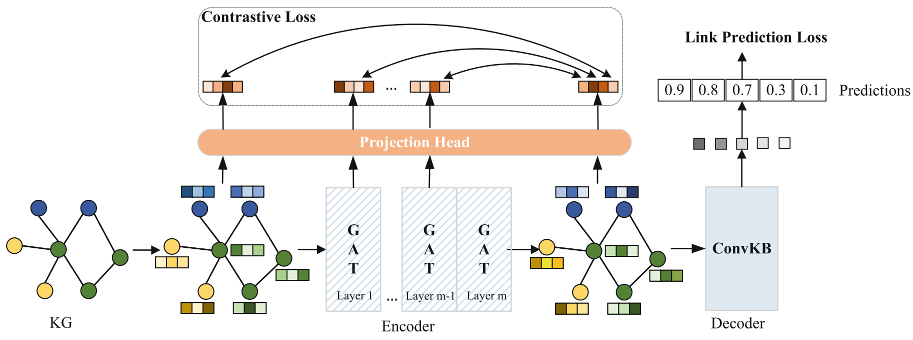


Fig. 1. Framework of the GCL-KGE model. We train the contrastive loss as a auxiliary task together with the link prediction loss.

3 Proposed Model

3.1 Overview

In this section, we describe our model which utilizes contrastive learning to learn the KG embedding. We present an encoder-decoder model called GCL-KGE in Fig. 1. The encoder learns knowledge graph embedding through the graph attention network to aggregate neighbor's information. And the decoder provides predictions for possible entities based on a triplet scoring function. We extend the existing model by introducing an auxiliary task to cope with interaction noise encountered in graph attention networks. First of all, we denote directed the KG as $G = (\nu, \xi)$ with nodes(entities) $v \in \nu$ and edges(relations) $r \in \xi$. Then we will introduce the details of the model.

3.2 Encoder

The neural network-based models encode entities and relations individually, ignoring the connections between the various triples in the knowledge graph. To capture the triple interaction information and graph structure information in the knowledge graph, we use a graph attention network to encode entities and relations based on work [1]. First, we obtain the initial embedding of entities and relations through a pre-trained model, using widely used embedding models. Then we place the embedding into the graph attention network to learn new representations. We learn new entity embedding h_i' in the form of the triple $t_k^{ij} = (h_i, h_k, h_j)$ where k is the relation link the entity i and entity j. A single GAT layer can be described as

$$a_{ijk} = softmax(LeakyRELU(W_2 b_{ijk})) \tag{1}$$

$$b_{ijk} = W_1[h_i : h_k : h_j] \tag{2}$$

where a_{ijk} is the attention score of the neighbor j. W_1 and W_2 are the linear transformation matrix mapping the initial embedding to a higher dimensional space. b_{ijk} is the embedding of a triple t_k^{ij}. Vector h_i,h_j and h_k denote embeddings of entities i, j and relation k respectively. Attention score is the importance of the neighbor j for entity i. Softmax is applied in Eq. (1) to compute the attention score.

In order to make the network capture more abundant neighbor information about various aspects, we use a multi-head attention mechanism to learn the embedding of entities. The formula shows the output of a layer:

$$h_i' = \sigma(\sum_{j\in\nu_i}\sum_{k\in\xi_{ij}} a_{ijk}b_{ijk}) \tag{3}$$

where ν_i denotes the neighbors of entity i and ξ_{ij} denotes the set of relations between entities i and j. The process of concatenating N attention heads is shown as follows.

$$h_i' = ||_{n=1}^{N} \sigma(\sum_{j\in\nu_i}\sum_{k\in\xi_{ij}} a_{ijk}^n b_{ijk}^n) \tag{4}$$

where $||$ represents concatation. σ represents a non-liner function. a_{ijk}^n is the normalized attention coefficients of the neighbor calculated in the n-th attention head.

In the final layer of the GAT, we employ averaging to get the final embedding of entities instead of the concatenation operation, as shown:

$$h_i' = \sigma(\frac{1}{N}\sum_{n=1}^{N}\sum_{j\in\nu_i}\sum_{k\in\xi_{ij}} a_{ijk}^n b_{ijk}^n) \tag{5}$$

We obtain the final entity embedding h_i' through the process described above. The graph attention network as the encoder of the whole model aggregates information about the surrounding neighbors into the entity's representation. In brief, an m-layer graph attention network module is able to gather information about the m-hop neighborhood.

3.3 Decoder and Score

The link prediction task is used to evaluate the effectiveness of our embeddings. We use ConvKB as the decoder of the GCL-KGE. Multiple filters are used to generate different feature graphs to capture global relations and transition characteristics between entities. We determine whether each triple (h, r, t) is a true triple by the scoring function.

$$f(t_{ij}^k) = (concat(g[h_i, h_k, h_j] * \Omega)) \cdot W \tag{6}$$

where W is a linear transformation matrix to score the triple. g is the activate function. Ω is the number of layers of convolutional filter layers and the * is a convolution operator.

3.4 Contrastive Learning

The encoder-optimized entity representations are scored in the decoder. We observe that there are some entities with similar representations which lead to incorrect predictions by the decoder. To make the model more sensitive to entity semantics, we adopt a contrastive learning approach: we treat pre-trained entities' embedding and the hidden state of the GAT as the positive instances for the entity i. We use h_i^+ to denote the representation of the positive instance of entity i in the set ν. Different entities in the same batch are used as negative instances in the set μ. Before calculating the contrastive loss, we map the entity representation to the same embedding space through the projection head layer. We adopt the contrastive loss, InfoNCE, about an instance i.

$$L_c = \sum_{i \in \nu} -log \frac{exp(sim(h_i, h_i^+))/\tau}{\sum_{j \in \mu} exp(sim(h_i, h_j))/\tau} \tag{7}$$

where τ is a temperature hyperparameter. sim is the similarity calculation function and we use dot product operations in our models. As shown in Fig. 1, our model uses the hidden states in the previous m-1 layers as positive instance representations of entities. They are semantically similar and more pure to the final output of the graph attention networks.

3.5 Training Objective

For the given knowledge graph, we train its embedding using the proposed model, the loss of our framework is :

$$L(h, r, t) = L_s + L_c \tag{8}$$

where L_c is the contrastive loss we introduced above. We train the GCl-KGE model using the Adam optimizer to minimize the loss function L_s. We use the L2 as the regularizer in our work.

$$\begin{gathered} L_s = \sum_{(h,r,t) \in \{G \cup G'\}} log(1 + exp(l_{(hrt)} \cdot f(h, r, t))) + \frac{\lambda}{2} \|w\|_2^2 \\ l_{(hrt)} = 1 \qquad for(hrt) \in G \\ l_{(hrt)} = -1 \qquad for(hrt) \in G' \end{gathered} \tag{9}$$

3.6 Theoretical Analyses

We discuss Eq. (7) to explain how contrastive learning in GCL-KGE make it work inspired. Contrastive learning performs meaningful gradient optimization to guide the embedding of entities. The gradient of the contrastive learning to the entity i is as follows [3]:

$$\begin{aligned}\frac{\partial L_c(h_i)}{\partial h_i} &= -\frac{\partial}{\partial h_i}(h_i \cdot h_i^+/\tau) + \frac{\partial}{\partial h_i} log \sum_{j \in N} exp(h_i \cdot h_j/\tau) \\ &= \frac{1}{\tau}\left\{-h_i^+ + \frac{\sum_{j\in\mu} h_j exp(h_i \cdot h_j/\tau)}{\sum_{j\in\mu} exp(h_i \cdot h_j/\tau)}\right\}\end{aligned} \tag{10}$$

where $L_c(h_i)$ is the gradient of a single entity i. Then we can derive the trend of change when entity i is updated:

$$\begin{aligned}h_i^{n+1} &= h_i^n - \gamma\frac{\partial L_c(h_i)}{\partial h_i} \\ &= h_i^n + \gamma\frac{h_i^+}{\tau} - \gamma\frac{\sum_{j\in\mu} h_j exp(h_i \cdot h_j/\tau)}{\sum_{j\in\mu} exp(h_i \cdot h_j/\tau)}\end{aligned} \tag{11}$$

in which h_i^n is the representation of entity i at this time step and h_i^{n+1} is the representation of entity i at the next time step. The Eq. (11) shows that h_i tends to update in the direction of h_i^+ with the weighted value $\frac{\gamma}{\tau}$. Entities optimally retain their content while gaining information about their neighbors. This solves the problem of the negative weight given by excessive redundancy when noise is added. The other side, update of h_i in a direction that is far from h_j with the weighted value $\gamma\frac{\sum_{j\in\mu} exp(h_i \cdot h_j/\tau)}{\sum_{j\in\mu} exp(h_i \cdot h_j/\tau)}$. In the same vector space, this way pulls away entities from their semantically similar entity representations.

4 Experiments

To evaluate the validity of our model and the usefulness of contrastive learning, we conducted a series of experiments and explained them in this section.

4.1 Experimental Setup

Datsets. To evaluate our model, we use four knowledge graph datasets: WN18RR [15], FB15k-237 [16], NELL-995 [17], and Alyawarra Kinship [18]. We show the setup of each dataset in Table 1. We need to specifically note that WN18RR and FB15k-237 are expanded from WN18 [6] and FB15k [19], respectively. Previous work [16] has shown that there is an inverse relationship in WN18 and FB15k resulting in test sets missing and further causing overfitting of the model. Therefore the researchers created two subsets of WN18RR and FB15k-237 to solve the problem.

Parameter Settings. We set the graph attention network with two layers in the model. The number of heads for multi-head attention to 2 and the last layer for both entity and relation embeddings to 200. The optimizer for the model uses the Adam optimizer with a learning rate of 0.001. We adjusted the optimal result of the hyperparameter τ to 1.0.

Table 1. Statistics of the datasets.

Dataset	Entities	Relations	Train	Valid	Test	Total
WN18RR	40,943	11	86,835	3,034	3,134	93,003
FB15K-237	14,541	237	272,115	17,535	20,466	310,116
NELL-995	75,492	200	149,678	543	3,992	154,213
Kinship	104	25	8,544	1,068	1,074	10,686

Comparable Methods. The baseline we choose are some of the most widely used knowledge graph embedding models, including DistMult [20], ComplEx [14], ConvE [16], TransE [6], ConvKB [5], R-GCN [8], ATTH [21] and KGE-CL [3].

4.2 Main Results

The evaluation metrics we use are MR, MRR and Hit@n. MR(Mean Rank) is a data of averaging the ranking positions of all correct triples in the sort MRR (Mean Reciprocal Rank) is the inverse of the ranking of all the results given by the standard answers. Hit@n is the proportion of correct entities in the top n ranking and we use Hit@1,Hit@3 and Hit@10.

Table 2. Link prediction results on WN18RR and NELL-995 datasets. We bold the best score in the table.

Methods	WN18RR				NELL-995			
	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
DistMult	0.444	0.412	0.47	0.504	0.485	0.401	0.524	0.61
ComplEx	0.449	0.409	0.469	0.53	0.482	0.399	0.528	0.606
ConvE	0.456	0.419	0.47	0.531	0.491	0.403	0.531	0.613
TransE	0.243	0.427	0.441	0.532	0.401	0.344	0.472	0.501
ConvKB	0.265	0.582	0.445	0.558	0.43	0.37	0.47	0.545
R-GCN	0.123	0.207	0.137	0.08	0.12	0.082	0.126	0.188
ATTH	0.466	0.419	0.484	0.551	–	–	–	–
KGE-CL	0.512	0.468	**0.531**	**0.597**	–	–	–	–
Our method	**0.522**	**0.477**	0.49	0.581	**0.541**	**0.456**	**0.596**	**0.698**

Referring to previous work, we test our model in a filtered setting that we remove some corrupt triples in the datasets. The experimental results of link prediction in all datasets are presented in Table 2 and Table 3. Our method is effective in all four datasets. Specifically, we achieve optimal results on the FB15K-237, NELL-995, and Kinship datasets. We also achieve comparable results to the optimal results on the WN18RR dataset. Compared to the tensor decomposition-based models and distance-based models, our model can preserve semantic information while preserving the structural information of the knowledge graph. It can expand the neighborhood information around the entity to encapsulate and generate a meaningful entity representation. At the same time, compared to the

Table 3. Link prediction results on FB15K-237 and Kinship datasets. We bold the best score in the table.

Methods	FB15K-237				Kinship			
	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
DistMult	0.281	0.199	0.301	0.446	0.516	0.367	0.581	0.867
ComplEx	0.278	0.194	0.297	0.45	0.823	0.733	0.899	0.971
ConvE	0.312	0.225	0.341	0.497	0.833	0.738	0.917	0.981
TransE	0.279	0.198	0.376	0.441	0.309	0.9	0.643	0.841
ConvKB	0.289	0.198	0.324	0.471	0.614	0.44	0.755	0.953
R-GCN	0.164	0.1	0.181	0.3	0.109	0.03	0.088	0.239
ATTH	0.324	0.236	0.354	0.501	–	–	–	–
KGE-CL	0.37	0.276	0.408	0.56	–	–	–	–
Our method	**0.513**	**0.435**	**0.551**	**0.657**	**0.907**	**0.878**	**0.941**	0.98

R-GCN, which also uses the graph neural network, our model incorporates contrastive learning as an auxiliary task to avoid the inclusion of noisy information while improving the effectiveness of entity embedding.

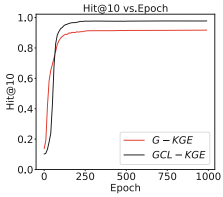
4.3 Ablation Experiments

Effect of Hyperparameter. We conduct the ablation experiment to evaluate the effect of parameter variations on the model. We confirm the importance of the hyperparameter τ in the contrastive loss for improving the efficiency. The model has the highest accuracy rates when the temperature parameter is 1.0 from the results of the ablation experiment in Table 4. The smaller the temperature, the more attention is paid to the difficult negative instances in the same batch. Part of the negative instances dominate the gradient optimization process, and the other negative samples do not work. Also, we evaluate the number of layers in projector head in contrastive learning. The model works best when the number of the neural network layers is 2.

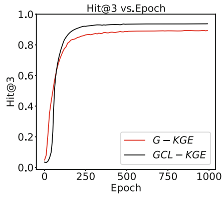
Table 4. MRR, Hit@1, Hit@3 and Hit@10 results of different value of τ and projection layers on the Kinship dataset.

Methods	Kinship			
	MRR	Hit@1	Hit@3	Hit@10
$\tau = 0.01$	0.886	0.845	0.909	0.967
$\tau = 0.1$	0.889	0.849	0.915	0.969
$\tau = 0.5$	0.891	**0.878**	0.936	0.977
$\tau = 1.0$	**0.907**	**0.878**	**0.941**	**0.98**
Projection-1 layer	0.863	0.814	0.892	0.958
Projection-2 layer	**0.907**	**0.878**	**0.941**	**0.98**
Projection-3 layer	0.875	0.818	0.91	0.96

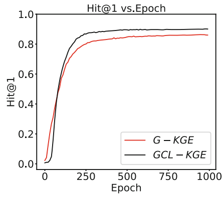
Effect of Contrastive Loss. We apply ablation experiments to the effect of contrastive learning. We remove the contrastive loss on the encoder-decoder framework called G-KGE. Figure 2 shows the comparison of the five metrics vs epoch on the kinship dataset. From the five subgraphs, we observe that GCL-KGE is significantly more effective than G-KGE without contrastive learning as an auxiliary task on the four metrics(Hit@10, Hit@3, Hit@1, MRR). (e) indicates the data on MR of GCL-KGE is lower than G-KGE. The figures also illustrate the improvement of contrastive learning on the GCL-KGE and the effectiveness of the choice of positive instances and negative instances.



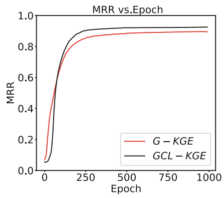
(a) Hit@10 vs.Epoch



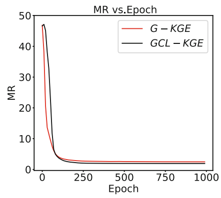
(b) Hit@3 vs.Epoch



(c) Hit@1 vs.Epoch



(d) MRR vs.Epoch



(e) MR vs.Epoch

Fig. 2. Hit@10, Hit@3, Hit@1, MRR and MR vs Epoch for GCL-KGE and the model without contrastive loss(G-KGE) on kinship dataset. GCL-KGE (black) represents the entire model.

5 Conclusion

In this work, we propose a knowledge graph embedding model combined with contrastive learning. We train the representation of entities and relations by graph attention networks, which aggregate graph structure information and multi-order neighbor semantic information. Then the triple scoring function in the ConvKB is used as a decoder for solving the link prediction task. In addition, we combine with contrastive learning as an auxiliary task to avoid the noise of graph attention networks. We propose a new method to construct positive instances which do not require data augmentation. The idea makes the entity embedding of the hidden layer function. Experimental results on four datasets demonstrate the effectiveness of our model.

In future work, we expect that contrastive learning can be applied more to knowledge graph embedding because it has been demonstrated to be helpful in representation learning in many studies. We hope that the development of self-supervised learning will be beneficial to solve the sparsity of knowledge graphs and improve the generality and transferability of knowledge graph embedding models.

Acknowledgements. This work was supported in part in part Key R & D project of Shandong Province 2019JZZY010129, and in part by the Shandong Provincial Social Science Planning Project under Award 19BJCJ51, Award 18CXWJ01, and Award 18BJYJ04.

References

1. Nathani, D., Chauhan, J., Sharma, C., Kaul, M.: Learning attention-based embeddings for relation prediction in knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4710–4723 (2019)
2. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
3. Xu, W., Luo, Z., Liu, W., Bian, J., Yin, J., Liu, T.Y.: KGE-CL: contrastive learning of knowledge graph embeddings. arXiv e-prints, arXiv-2112 (2021)
4. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2021, pp. 2069–2080 (2021)
5. Dai Quoc Nguyen, T.D.N., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. In: Proceedings of NAACL-HLT, pp. 327–333 (2018)
6. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
7. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Icml (2011)
8. Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: ESWC (2018)
9. Shang, C., Tang, Y., Huang, J., Bi, J., He, X., Zhou, B.: End-to-end structure-aware convolutional networks for knowledge base completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 3060–3067 (2019)
10. Chen, M., Wei, F., Li, C., Cai, D.: Frame-wise action representations for long videos via sequence contrastive learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13801–13810 (2022)
11. Chen, X., He, K.: Exploring simple Siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758 (2021)
12. Meng, Y., Xiong, C., Bajaj, P., Bennett, P., Han, J., Song, X.: Coco-LM: correcting and contrasting text sequences for language model pretraining. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
13. Wu, Z., Wang, S., Gu, J., Khabsa, M., Sun, F., Ma, H.: Clear: contrastive learning for sentence representation. arXiv preprint: arXiv:2012.15466 (2020)

14. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080. PMLR (2016)
15. Toutanova, K.: Observed versus latent features for knowledge base and text inference. ACL-IJCNLP **2015**, 57 (2015)
16. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1 (2018)
17. Xiong, W., Hoang, T., Wang, W.Y.: DeepPath: a reinforcement learning method for knowledge graph reasoning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 564–573 (2017)
18. Lin, X. V., Socher, R., Xiong, C.: Multi-hop knowledge graph reasoning with reward shaping. In: EMNLP (2018)
19. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Twenty-fifth AAAI Conference on Artificial Intelligence (2011)
20. Yang, B., Yih, S. W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (2015)
21. Chami, I., Wolf, A., Juan, D.C., Sala, F., Ravi, S., Ré, C.: Low-dimensional hyperbolic knowledge graph embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6901–6914 (2020)
22. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI Conference on Artificial Intelligence (2017)
23. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: ACL (2015)