Bootstrapped Knowledge Graph Embedding based on Neighbor Expansion

Jun Seon Kim
School of Computing, KAIST
Daejeon, Republic of Korea
junseon.kim@kaist.ac.kr

Seong Jin Ahn
School of Computing, KAIST
Daejeon, Republic of Korea
sja1015@kaist.ac.kr

Myoung Ho Kim
School of Computing, KAIST
Daejeon, Republic of Korea
mhkim@kaist.ac.kr

ABSTRACT

Most Knowledge Graph(KG) embedding models require negative sampling to learn the representations of KG by discriminating the differences between positive and negative triples. Knowledge representation learning tasks such as link prediction are heavily influenced by the quality of negative samples. Despite many attempts, generating high-quality negative samples remains a challenge. In this paper, we propose a novel framework, Bootstrapped Knowledge graph Embedding based on Neighbor Expansion (BKENE), which learns representations of KG without using negative samples. Our model avoids using augmentation methods that can alter the semantic information when creating the two semantically similar views of KG. In particular, we generate an alternative view of KG by aggregating the information of the expanded neighbor of each node with multi-hop relation. Experimental results show that our BKENE outperforms the state-of-the-art methods for link prediction tasks.

CCS CONCEPTS

• **Computing methodologies → Knowledge representation and reasoning**; *Neural networks.*

KEYWORDS

Knowledge Graph Embedding, Graph Embedding, Self-Supervised Learning, Graph Neural Networks, Link Prediction

ACM Reference Format:

Jun Seon Kim, Seong Jin Ahn, and Myoung Ho Kim. 2022. Bootstrapped Knowledge Graph Embedding based on Neighbor Expansion. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557555

1 INTRODUCTION

A knowledge graph(KG) is a multi-relational graph structure that represents real-world knowledge in triple forms (head entity, relation, tail entity). KGs are utilized in a variety of applications, including question answering[8, 10], link prediction[7, 15], and machine reading comprehension[10, 26]. However, KGs suffer from missing relations, and to overcome this problem, researchers have focused on the link prediction task to predict the missing facts in KGs. The link prediction methods lead to a Knowledge Graph Embedding (KGE) that learns the representation of relations and entities to represent the inherent structure of the KG. In recent years, various knowledge graph embedding models are proposed, from conventional methods such as TransE[3], RESCAL[16] to neural network-based methods such as ConvE[7] and KBAT[15].

Most KGE models define their own scoring functions that give a score on the plausibility of a given triple. KGE learns the representations of KG by ranking the scores of observed triples (positive) higher than those of unobserved triples (negative)[17]. Therefore, KGE models require a lot of negative samples for training. The most common negative sampling method is uniform sampling[3, 14], which replaces the head or tail entity of a positive triple with uniformly selected ones from the KG entity set $\mathcal{E}$. These randomly generated negative samples often result in poor performance of KGE. Thus, generating high-quality negative samples is crucial to knowledge graph embedding. In order to obtain high-quality negative samples, other approaches such as GAN-based sampling[4, 5, 19], cache-based methods[29], and structure-aware sampling[1] have been proposed. However, generating high-quality negative samples of KGE still remains a challenge.

To deal with the difficulty of acquiring high-quality negative samples, there have been attempts to learn representations using only positive samples in many domains, such as images and graphs. In the vision domain, a self-supervised learning method, BYOL[9] shows competitive performance compared with the SOTA methods without using negative samples. BGRL[21] employs a BYOL approach to graph representation learning. It learns node presentation by minimizing cosine similarity between node presentations derived from two augmented views of the graph. Since KGs differ from general graphs in several ways, applying the BGRL method to KGEs is challenging. The BGRL and other augmentation-based contrastive models[6, 25, 30] use simple augmentation methods, such as edge dropping and node feature masking. When KGs are augmented with edge dropping, however, it becomes more difficult to capture semantic information, such as multi-hop relation. Furthermore, simply minimizing node similarity in BGRL is not sufficient to capture the relation information that is vital to KGs. Accordingly, to apply the BYOL approach to the KGs while retaining their structural attributes, 1)we require two semantically similar views of KGs that are not augmented using current methods, and also 2) we need to include relation information within KGs.

In this paper, we propose a novel framework, Bootstrapped Knowledge graph Embedding based on Neighbor Expansion (BKENE), which learns knowledge graph representations without using negative samples. Since simple augmentation may lead to information

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557555

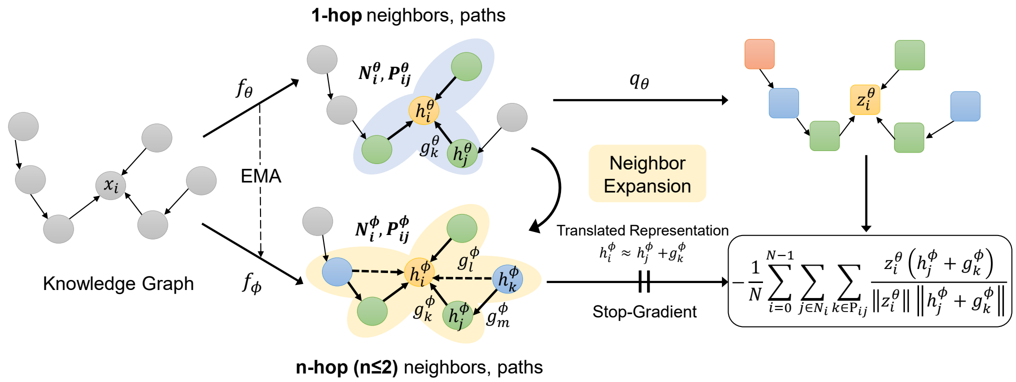

Figure 1: The architecture of BKENE framework. Two encoders f_θ, f_ϕ generate embeddings for all nodes of KG. To obtain two different representations h_i^θ, h_i^ϕ of node x_i, we expand the set of original neighbors(N_i^θ) to n-hop neighbors (N_i^ϕ). Dashed arrows represent newly aggregated paths as a result of neighbor expansion. A predictor q_θ only projects the online representation. Loss function is computed for z_i^θ along with the $\|h_j^\phi + g_k^\phi\|$, updating only f_θ with gradients, while f_ϕ is updated by an EMA of f_θ.

loss and semantic changes in KGs[12], we generate two different views of KGs by defining the neighbors of each node differently in two encoders. One encoder aggregates the information of only directly connected nodes and relations, and another aggregates the information of expanded neighbors which include multi-hop neighbors. Then given the two semantic related views, we train the node representation of the first encoder to predict the translated representation of the second encoder by using the newly proposed objective function. According to our experimental findings, BKENE is effective for link prediction tasks without negative sampling, and it outperforms existing state-of-the-art KGE models. The following are our contributions:

- We propose BKENE, a self-supervised knowledge graph embedding method that does not require negative sampling.
- To obtain alternative views of a knowledge graph by adopting the BYOL framework, we aggregate different neighbor information by neighbor expansion with relation-aware objective functions.
- On link prediction tasks, our proposed method outperforms existing SOTA KGE models.

2 OUR APPROACH

2.1 Preliminaries

Knowledge Graph A knowledge graph is represented by $\mathcal{G} = \{(h,r,t)|h \in \mathcal{E}, r \in \mathcal{R}, t \in \mathcal{E}\}$, where $\mathcal{E}$ and $\mathcal{R}$ indicate entities and relations, respectively. Each triple (h,r,t) indicates that the head entity h and the tail entity t have a relation r between them.

BYOL : Bootstrap Your Own Latent We apply the BYOL[9] approach to KG to overcome the challenge of getting high-quality negative samples. By bootstrapping the delayed latent representation, the BYOL provides an effective way to learn the representations of images without using negative samples.

There are two encoders in BYOL: the online encoder f_θ and the target encoder f_ϕ. Both encoders have the same structure, but with different parameters. Given the image x, BYOL produces the two augmented views of images (v, v') by applying each augmentation method, and these augmented images are fed into each encoder, respectively. Online and target encoders generate their representations $y_\theta = f_\theta(v)$, $y'_\phi = f_\phi(v')$ and projections $z_\theta = g_\theta(y_\theta)$, $z'_\phi = g_\phi(y'_\phi)$, successively. However, only the online encoder applies a prediction $q_\theta(z_\theta)$, which makes an asymmetric between the online and the target network. Finally, $\mathcal{L}_{\theta,\phi} = \|\overline{q_\theta}(z_\theta) - \overline{z'}_\phi\|_2^2$ is an loss function, where $\overline{q_\theta}(z_\theta)$ and $\overline{z'}_\phi$ is the l2-normalized term of $q_\theta(z_\theta)$ and z'_ϕ. To symmetrize the loss, it computes the $\tilde{\mathcal{L}}_{\theta,\phi}$ by exchanging the augmented input images of each encoder, i.e., v' to online encoder f_θ and v to the target encoder f_ϕ. The final objective function is defined as $\mathcal{L}^{BYOL}_{\theta,\phi} = \mathcal{L}_{\theta,\phi} + \tilde{\mathcal{L}}_{\theta,\phi}$ and it is minimized by a stochastic optimization at each iteration steps, with respect to parameter θ, while the ϕ is updated by an exponential moving average of θ.

2.2 Bootstrapped Knowledge graph Embedding based on Neighbor Expansion(BKENE)

We propose a novel self-supervised framework, Bootstrapped Knowledge graph Embedding based on Neighbor Expansion(BKENE). The overall BKENE framework is shown in Figure 1.

2.2.1 Encoder with Bootstrapped Embeddings.

Instead of using augmentation, to get the two semantically similar views, we define a Graph Neural Network (GNN)-based encoders : $f_\theta(\cdot)$, $f_\phi(\cdot)$, which have same architecture with different sets of parameters. Each encoder gets the original graph $\mathcal{G}$ as input and computes the online and target representation of each node h_i^θ h_i^ϕ, respectively, for $v_i \in \mathcal{E}$. We differently define the set of neighbors in each encoder to obtain two different representations for each node. Encoder f_θ aggregates only the information from neighbor nodes directly connected to a given node. While, the other encoder f_ϕ includes not only directly connected nodes but also multi-hop neighbor nodes and their associated relations, using multi-hop relation representation. We also assign importance to the triples associated with a given node by using an attention mechanism.

Relation Path-based Triple Representation To represent the semantics of a multi-hop relation in KG, we define a path $p = (r_1, r_2, ..., r_n)$ that is composed of multiple relations. We compute the path embedding using 3 different semantic composition operations proposed by [13]: Addition(Add) : $p = r_1 + ... + r_n$, Multiplication(Mult) : $p = r_1 \cdot ... \cdot r_n$, Recurrent Neural Network(RNN) : $p = c_n, c_n = f(W[c_{n-1}; r_n])$,
$(c_1 = r_1)$

To learn the new embedding of entity e_i that aggregates the information from neighbors, representations of triples related to e_i are also learned. To obtain the vector representation of triples, we concatenate the initialized entity and relation embedding of each triple. We define a path-based triple t_{ij}^k, and its embedding c_{ijk}.

$$t_{ij}^k = (e_i, p_k, e_j), p_k = (r_1, r_2, ..., r_k) \tag{1}$$

$$c_{ijk} = W_1 \left[h_i \parallel h_j \parallel g_k \right] \tag{2}$$

,where c_{ijk}, h_i, h_j, g_k represent the vector embedding of t_{ij}^k, e_i, e_j, p_k, respectively, and W_1 denotes the linear transformation.

Expanded Neighbor Aggregation with Attention In KGs, there can be various relation paths between e_i and e_j. Thus, we adapt the graph attention layer[15] so that we can assign a different attention score α_{ijk} to each path-based triple c_{ijk}. To calculate the attention values we use the softmax function on the output of equation 2 and applied a linear transformation W_2 and the LeakyRelu function.

$$\alpha_{ijk} = \frac{exp(LeakyReLU(c_{ijk}W_2))}{\sum_{j \in N_i} \sum_{k \in P_{ij}} exp(LeakyReLU(c_{ijk}W_2))} \tag{3}$$

,where N_i represents the set of neighbors of an entity e_i and P_{ij} denotes the set of relation paths between an entity e_i and e_j.

Based on our encoder structure, each encoder $f_\theta(\cdot)$, $f_\phi(\cdot)$, has different N_i and P_{ij} represented as N_i^θ, N_i^ϕ, P_{ij}^θ, P_{ij}^ϕ, respectively. For encoder $f_\theta(\cdot)$, we define N_i^θ as a set of nodes directly connected to e_i. By the definition of the N_i^θ, P_{ij}^θ can be defined as a set of 1-hop relations between node e_i and e_j. While, in the encoder $f_\phi(\cdot)$, we define the N_i^ϕ as a set of nodes in an n-hop relationship with node e_i. Following by N_i^ϕ, P_{ij}^ϕ is denoted as an set of relation paths consisting of one or more connected relations.

$$h^{\theta\prime}_i = \sigma\left(\sum_{j \in N_i^\theta} \sum_{k \in P_{ij}^\theta} \alpha_{ijk} c_{ijk} \right), h^{\phi\prime}_i = \sigma\left(\sum_{j \in N_i^\phi} \sum_{k \in P_{ij}^\phi} \alpha_{ijk} c_{ijk} \right) \tag{4}$$

Finally, to get node representation, we aggregate the neighbor information according to their attention scores and apply the non-linearity activation as equation 4. Instead of using arbitrary graph augmentation, we obtain two different representations for each node created by two different encoders, one of which uses neighbor expansion.

2.2.2 Relation-Aware Objective Function.

We propose an objective function that incorporates the relation information of KG by representing the relation as a translation between two entities in the vector space. Our objective function aims to minimize the cosine similarity between the query entity e_i and its translated representation $\|e_j + p_k\|$ based on path-based triple $t_{ij}^k = (e_i, p_k, e_j)$ as we define on section 2.2.1.

$$\mathcal{L}_{\theta,\phi} = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{j \in N_i} \sum_{k \in P_{ij}} \frac{z_i^\theta (h_j^\phi + g_k^\phi)}{\|z_i^\theta\| \|h_j^\phi + g_k^\phi\|} \tag{5}$$

,where $q_\theta(\cdot)$ is a predictor network and $z_i^\theta = q_\theta(h_i^\theta)$ is the prediction of the online embedding h_i^θ. Also, N_i, P_{ij} denote the set of neighbors for an entity e_i and the set of relation paths between head e_i and tail e_j entities, respectively. To symmetrize this loss function, we also calculate the $\tilde{\mathcal{L}}_{\theta,\phi}$, by swapping the definition of neighbor of the two encoders. Therefore, the final object of our model is to minimize the $\mathcal{L}_{\theta,\phi}^{BKENE} = \mathcal{L}_{\theta,\phi} + \tilde{\mathcal{L}}_{\theta,\phi}$.

Updating Parameters In our method, the online network is updated based on the gradients of the objective function(Equation 5) with respect only to online parameters θ. While the target network, with a set of parameters ϕ, is updated as an exponential moving average of θ, where η, τ is the learning rate of the online network and the decaying rate, respectively.

$$\theta \leftarrow optimizer\left(\theta, \nabla_\theta \mathcal{L}_{\theta,\phi}^{BKENE}, \eta\right), \phi \leftarrow \tau\phi + (1-\tau)\theta \tag{6}$$

2.2.3 Decoder.

Among many existing knowledge graph completion methods, we adapt the ConvE[7] as a decoder, which is considered robust under the different evaluation protocols[20]. ConvE[7] uses 2D convolutions on KGE based on the head and relation pair query. The triple (h, r, t) is scored as:

$$f(e_h, e_t) = f(vec(f([\bar{e_h} : \bar{r_r}] * \omega))W)e_t \tag{7}$$

,where $r_r \in \mathbb{R}^k$ is a relation parameter related to r, $\bar{e_h}, \bar{e_t}$ denote a 2D shaping of e_h, e_t, respectively, with ω that denotes the convolution filter and linear transformation matrix $W \in R^{cmn \times k}$, where c denote the number of 2D feature maps with dimension m and n, and k denotes the entity embedding dimension.

3 EXPERIMENTS

3.1 Datasets

We conducted experiments for the KG link prediction task on the two most popular public datasets: FB15K-237 and WN18RR. Table 1 summarizes the statistical information of the two datasets.

Table 1: Dataset Statistics

Datasets	♯Entities	♯Relations	♯Training	♯Validation	♯ Test	Mean in-degree
FB15K-237	14,541	237	272,115	17,535	20,466	18.71
WN18RR	40,943	11	86,835	3,034	3,134	2.12

3.2 Experimental Setup

3.2.1 Implement Details. The embeddings of the entities and relations in KG are initialized with 100 dimensions of pre-trained embeddings used in [20]. The model is trained by a 2 layer graph attention-based encoder with 2 multi-head. After performing a grid search for hyperparameters, we use PReLu activation on each layer and set the hidden dim of embeddings as 200, and the learning rate and decaying rate are set as $\eta = 0.001$, $\tau = 0.99$, respectively. We adopt Adam optimization to our models with a full-batch setting.

	FB15K-237					WN18RR				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
TransE[3]	323	0.279	0.198	0.376	0.441	<u>2300</u>	0.243	0.044	0.441	0.532
DistMult[28]	512	0.281	0.199	0.301	0.446	7000	0.444	0.412	0.470	0.504
ComplEx[22]	339	0.247	0.158	0.257	0.428	5261	0.440	0.440	0.460	0.510
ConvE[7]	244	0.325	0.237	0.356	0.501	4187	0.430	0.400	0.440	0.520
SACN[18]	_	0.350	0.260	<u>0.390</u>	<u>0.540</u>	_	0.470	0.430	0.480	0.540
ConvR[11]	_	0.350	0.261	0.385	0.528	_	0.475	0.443	0.489	0.537
MuRE[2]	_	0.336	0.245	0.370	0.521	_	0.475	0.436	0.487	0.554
RotaE[19]	<u>177</u>	0.338	0.241	0.375	0.533	3340	0.476	0.428	0.492	0.571
InteracE[24]	**172**	0.354	<u>0.263</u>	_	0.535	3340	0.476	0.428	_	0.528
ReInceptionE[27]	173	0.349	_	_	0.528	**1894**	<u>0.483</u>	_	_	<u>0.582</u>
CompGCN[23]	197	<u>0.355</u>	0.254	<u>0.390</u>	0.535	3533	0.479	<u>0.443</u>	<u>0.493</u>	0.546
KBGAN[5]*	_	0.293	_	_	0.466	_	0.181	_	_	0.432
NSCaching[29]*	_	0.299	_	_	0.476	_	0.200	_	_	0.478
SANS[1]*	_	0.298	_	_	0.485	_	0.231	_	_	0.534
BKENE(ours)	240	**0.381**	**0.298**	**0.429**	**0.570**	3746	**0.484**	**0.445**	**0.512**	**0.584**

Table 2: Results of the Link Prediction task on FB15K-237 and WN18RR test datasets. * denotes that each result is taken from [1], while the others directly come from the corresponding papers. Best results are highlighted in bold, whereas second-best results are <u>underlined</u>.

We train our encoder with 3200 epochs in FB15K-237 and 3600 epochs in WN18RR, while the decoder with at least 200 epochs.

3.2.2 Evaluation Metrics. The link prediction task aims to predict the elements that are missing from the test triples. A ranking process is used to evaluate the results from the link prediction task. Following the fair evaluation environment used in [20], we perform our model in a "filtered setting" with the random protocol. To evaluate our model, we utilize ranking metrics including Mean Rank (MR), Mean Reciprocal Rank (MRR), and the proportion of correct entities ranking in the top N, Hits@N(N=1, 3, 10). In general, a higher MRR, Hits@N, and a lower MR imply better performance.

3.3 Results and Analysis

We compare the performance of our BKENE on link prediction tasks with various KGE models from conventional models (TransE[3], DistMult[28], ComplEx[22], ConvE [7]) to current state-of-the-art models (SACN[18], ConvR[11], MuRE[2],InteractE[24], RoatE[19] ReInceptionE[27], and CompGCN[23]). We also compared our BKENE with the models presenting the negative sampling method applied to the TransE scoring function : KBGAN[5], NSCaching[29], SANS[1].

From Table 2, we can see that our proposed BKENE outperforms the existing SOTA KGE models. BKENE also shows the best results in FB15K-237 and WN18RR with MRR and Hits@1, 3, 10. Also, compared with negative sampling methods (KBGAN[5], NSCaching[29], SANS[1]), our model achieves significant improvements at MRR and Hits@10. This implies that our novel self-supervised framework can capture the semantics of KG effectively with only positive samples. Especially, in FB15K-237, BKENE shows $0.570 - 0.540 = 0.03$ improvements compared with the second-best performance model, while it shows $0.584 - 0.582 = 0.002$ improvements on WN18RR. It shows that BKENE is more effective on FB15K-237, which includes more relations and is much denser than WN18RR. We attribute the excellence of our model to its ability to capture the semantics of KG, by expanding the range of neighbors, rather than using arbitrarily generated negative samples. Also, the model can learn more latent features by using a decoder.

Table 3: Performance of BKENE with using different semantic composition operators.

	FB15K-237		WN18RR	
	MRR	Hits@10	MRR	Hits@10
BKENE(Add, 2-hop)	**0.381**	**0.570**	**0.484**	**0.584**
BKENE(Mult, 2-hop)	0.379	0.563	0.471	0.558
BKENE(RNN, 2-hop)	0.361	0.553	0.464	0.543
BKENE(Add, 3-hop)	0.378	0.565	0.473	0.561

The performances of each BKENE, implementing a relation path using three different semantic composition operators, are presented in Table 3. In both MRR and Hits@10, the addition operation outperforms the other two operations. Our models appear to benefit more from the addition operation. We also compare a BKENE applying 2-hop relation with a BKENE applying 3-hop relation. We found that the performance was degraded when the 3-hop relation path is applied, which indicates that a long relation path can contain unnecessary information in KGE.

4 CONCLUSION

We have introduced BKENE, a novel self-supervised knowledge graph embedding method. BKENE avoids the need for negative sampling, which can significantly impact the performance of KGE. Our BKENE generates another view of KG by defining the range of neighboring nodes differently by using neighbor expansion, instead of arbitrary augmentation. This model also utilizes relation information to minimize cosine similarity between representation of nodes and translated representations of these nodes. In WN18RR and FB15K-237 datasets, our model outperforms existing models on knowledge graph link prediction.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant (No. 2020R1A2C1004032), and the Bio-Synergy Research Project of the MSIT (No. 2013M3A9C4078137), and the ITRC support program (IITP-2020-2020-0-01795) supervised by the IITP.

REFERENCES

[1] Kian Ahrabian, Aarash Feizi, Yasmin Salehi, William L Hamilton, and Avishek Joey Bose. Structure aware negative sampling in knowledge graphs. *arXiv preprint arXiv:2009.11355*, 2020.

[2] Ivana Balazevic, Carl Allen, and Timothy Hospedales. Multi-relational poincaré graph embeddings. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[4] Avishek Joey Bose, Huan Ling, and Yanshuai Cao. Adversarial contrastive estimation. *arXiv preprint arXiv:1805.03642*, 2018.

[5] Liwei Cai and William Yang Wang. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071*, 2017.

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[8] Georg Gottlob and Christoph Koch. Logic-based web information extraction. *ACM SIGMOD Record*, 33(2):87–94, 2004.

[9] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[10] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231, 2017.

[11] Xiaotian Jiang, Quan Wang, and Bin Wang. Adaptive convolution for multi-relational learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 978–987, 2019.

[12] Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. *arXiv preprint arXiv:2112.02472*, 2021.

[13] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*, 2015.

[14] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[15] Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. Learning attention-based embeddings for relation prediction in knowledge graphs. *arXiv preprint arXiv:1906.01195*, 2019.

[16] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, 2011.

[17] Jing Qian. Understanding negative sampling in knowledge graph embedding. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 12(1), 2021.

[18] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3060–3067, 2019.

[19] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

[20] Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, and Yiming Yang. A re-evaluation of knowledge graph completion methods. *arXiv preprint arXiv:1911.03903*, 2019.

[21] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

[22] Théo Trouillon, Christopher R Dance, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *arXiv preprint arXiv:1702.06879*, 2017.

[23] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.

[24] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3009–3016, 2020.

[25] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (Poster)*, 2(3):4, 2019.

[26] Dirk Weissenborn, Pasquale Minervini, Tim Dettmers, Isabelle Augenstein, Johannes Welbl, Tim Rocktäschel, Matko Bošnjak, Jeff Mitchell, Thomas Demeester, Pontus Stenetorp, et al. Jack the reader-a machine reading framework. *arXiv preprint arXiv:1806.08727*, 2018.

[27] Zhiwen Xie, Guangyou Zhou, Jin Liu, and Xiangji Huang. Reinceptione: relation-aware inception network with joint local-global structural information for knowledge graph embedding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5929–5939, 2020.

[28] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[29] Yongqi Zhang, Quanming Yao, Yingxia Shao, and Lei Chen. Nscaching: simple and efficient negative sampling for knowledge graph embedding. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 614–625. IEEE, 2019.

[30] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.