ResConvE: Deeper Convolution-Based Knowledge Graph Embeddings

Yongxu Long, Zihan Qiu, Dongyang Zheng, Zhengyang Wu, Jianguo Li, and Yong Tang(✉)

School of Computer Science, South China Normal University, Guangzhou 510631, Guangdong, China
{longyongxu,ytang}@m.scnu.edu.cn

Abstract. Link prediction on knowledge graphs (KGs) is an effective way to address their incompleteness. ConvE and InteractE have introduced CNN to this task and achieved excellent performance, but their model uses only a single 2D convolutional layer. Instead, we think that the network should go deeper. In this case, we propose the ResConvE model, which takes reference from the application of residual networks in computer vision, and deepens the neural network, and applies a skip connection to alleviate the gradient explosion and gradient disappearance caused by the deepening of the network layers. We also introduce the SKG-course dataset from Scholat for experiments. Through extensive experiments, we find that ResConvE performs well on some datasets, which proves that the idea of this method has better performance than baselines. Moreover, we also design controlled experiments setting different depths of ResConvE on FB15k and SKG-course to demonstrate that deepening the number of network layers within a certain range does help in performance improvement on different datasets.

Keywords: Knowledge graph embedding · Residual network · Knowledge graph · SCHOLAT · Link prediction

1 Introduction

Knowledge Graphs (KGs) are structured knowledge bases that are constructed by facts. One fact in KGs includes subject s, relation r, and object o, i.e. triplet (s, r, o), which means KGs are the collections of such triplets. Since Google announced its Knowledge Graph in 2012, many KGs such as WordNet [13], YAGO [21], Freebase [2], and SKG (Scholat Knowledge Graph) keep coming these years. They find various applications in quantities of area, for example, relation extraction, search, analytics, recommendation, and question answering [27,30].

However, the main problem faced by knowledge graphs when applied is incompleteness [5]. In particular, links in the KGs are missing, for example, 71% of users

This work was supported in part by the National Natural Science Foundation of China under Grant U1811263 and Grant 61772211.

Y. Sun et al. (Eds.): ChineseCSCW 2021, CCIS 1492, pp. 162–172, 2022.
https://doi.org/10.1007/978-981-19-4549-6_13

in Freebase are missing birthday information and 75% are missing nationality information [5]. To solve this problem, methods such as TransE [3] and TransH [28] are based on using existing subjects and relations in KGs, performing embedding operations to map them into a vector space, and making predictions, while the model learns parameters and constantly optimizes the score function.

In fact, however, real-world knowledge graphs are so large that link prediction [12] for such knowledge graphs requires not only the number of parameters to be considered, but also the computational cost. For shallow models such as TransE [3], TransH [28] and DisMult [29], the best way to make improvements is to increase the size of the embedding matrix, which may most likely result in an excessive number of parameters for large knowledge graphs [4]. ConvE [4] performs very well in several tasks, but the structure of the model is too shallow for CNNs and can not make sure that the features of the KG can be fully learned. Therefore, inspired by the shortcut idea in ResNets [8], we propose the ResConvE model to embed the KGs.

ResConvE adds several more convolutional layers to ConvE. However, in view of the problem of gradient disappearance and gradient explosion [1,6,7] caused by deeper networks, ResConvE uses a skip connection mechanisms, so that the model can achieve good results even if it is very deep.

Our contributions are as follows:

1. Inspired by ConvE, which treats entities and relationships as "images", we propose ResConvE, the first application of the idea of ResNets to link prediction models based on knowledge graph embedding, which deepens the neural network without losing the original ability to extract features, providing a new idea for link prediction model building.
2. ResConvE was evaluated on various link prediction datasets and proved to be more effective in most of the datasets. Meanwhile, we explored how effective deepening the neural network model was in improving performance.
3. The SKG-course dataset was introduced for link prediction tasks.

2 Related Work

Since the introduction of the TransE [3] model, a variety of link prediction models have emerged. Early models use the translation objective as the score function including the TransE [3] and TransH [28], and the DisMult [29] model, which is based on a bilinear diagonal, but although their models are effective, they do not deepen the neural network, which makes them less effective than our model.

The introduction of ConvE [4] provides a new way of thinking about link prediction models, and ConvE proposes to use 2D convolutional layers to build a neural network model for link prediction. InteractE [26] improves on ConvE: when entity vectors are spliced with relational vectors, the model first rearranges the concatenated vectors according to rules and then feeds them into a 2D convolutional for training.

ConvE has been very successful in introducing 2D convolutional layers. InteractE, based on ConvE, has improved on this by doing vectors processing before

feeding in 2D convolutional layers, with good results. However, both of these end up using only one 2D convolutional layer, and there is no discussion of the effects of deepening the model in either of these articles. However, we believe that the structure is too simple and the network is not deep enough for CNNs, of which the advantage is feature extraction. Meanwhile, we know that making a simple stack of neural network layers to deepen it can lead to gradient explosion or gradient disappearance, hindering the emergence of loss function convergence and even making the accuracy degrade rapidly after reaching saturation. Inspired by the structure of the ResNets [8] network, we introduced a Shortcut mechanism in ResConvE to solve this problem. Shortcut refers to the back-propagation of the model by skipping one or more layers of connections and adding the data directly to the output of the mainstem, which is a stack of network layers, during forward propagation.

3 Background

Knowledge Graph: A knowledge graph $\mathcal{G}$ is a collection of triplets (s, r, o), consisting of relations r, subjects s and objects o. Figure 1 illustrates the structure of a triplet.

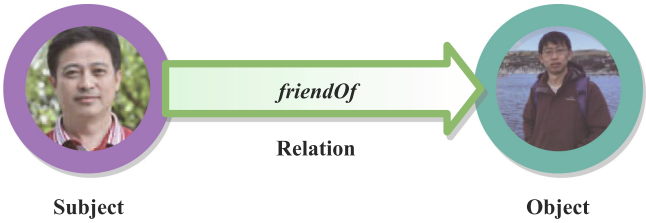

Fig. 1. A relation, a subject, and an object form a triplet.

Knowledge Graph Link Prediction: The main task of the link prediction in KGs is to make use of the existing facts in KGs to predict the new ones, which means we need the model to learn a score function ψ with an input triplet (s, r, o) whose score depends on the likelihood of the fact being true, to which is proportional.

Entities and relations will be encoded in most of the existing KGs embedding approaches. The validity of the output triplets will then be measured by a defined score function. Some score functions for existing models are presented in Table 1. Once the score function is defined, the model learns based on the inputs and outputs, thus continuously optimizing the model parameters.

ConvE: ConvE introduces 2D convolution into the model for KG link prediction. convE feeds entities and relationships into the 2D convolution layer after embedding them. The score function ψ for ConvE is given as follow:

$$\psi = f_a(vec(f_a(cat(\bar{emb}_s, \bar{emb}_r) \star \omega))W)emb_o \tag{1}$$

where $\bar{emb}_s$ and $\bar{emb}_r$ represent the tensors after 2D reshaping of the embedding matrices of subject emb_s and relation emb_r, while $\star$ denotes the convolution operation. e_o represents the embedding output, normally a matrix, of the object, and W represents the weight matrix to be learned. f_a refers to the activation function. 2D reshaping is considered to be useful for learning the representation of entities and relations.

Table 1. Some score functions for existing models. $\star$ represents convolution.

Model	Score function
TransE [3]	$\|emb_s + emb_r - emb_o\|_p$
DisMult [29]	$\langle emb_s, emb_r, emb_o \rangle$
ConvE [4]	$f_a(vec(f_a(cat(\bar{emb}_s, \bar{emb}_r) \star \omega))W)emb_o$
InteractE [26]	$f_a(vec(f_a(Perm(\mathcal{P}_k) \star \omega))W)emb_o$
ResConvE	$f_a(vec(f_a(cat(\bar{emb}_s, \bar{emb}_r)) \star \omega + \sum f_a(cat(\bar{emb}_s, \bar{emb}_r)) \star \omega')W)emb_o$

4 ResConvE

4.1 Overview

ConvE [4] indicates that using 2D convolution does boost the expressiveness of the model. The expressive ability of ConvE is further enhanced by reconstruction of the entity-relationship embedding before feeding into the convolutional layer for computation in InteractE [26]. From the experience of CNN in computer vision, we believe that a deeper network structure is conducive to capturing richer entity attributes and relationship features. So in order to extend the approach to capture entity-relationship features, ResConvE proposes the following two ideas:

1. **Deepening the neural network:** In contrast to ConvE and InteractE, which use only a single layer of 2D convolutional layers, ResConvE uses multiple convolutional layers to deepen the neural network, which is normally an important trick in the field of computer vision [18–20,22]. Inspired by ConvE's introduction of CNNs to the knowledge graph embedding task, we build on this approach to deepen the network, to extract features of entities and relationships better.
2. **Shortcut:** Many theories and practices have shown that if a neural network is only deepened, the gradients will eventually explode or vanish [7,8]. Inspired by ResNets [8], ResConvE introduces a shortcut mechanism, which allows the deepening of the neural network without compromising the model's capabilities.

4.2 Detail

The overall architecture of ResConvE is shown in Fig. 2. ResConvE learns a vector of dimension d to represent entities or relationships in KG. In forward propagation, the data input to ResConvE will be divided into two paths after the embedding operation, which are mainstem and shortcut, where the mainstem will have multiple convolutional layers, and the shortcut, with two 2D convolutional layers, makes sure that the model will not damage the original capability in the process of deepening.

Before Divided into Two Paths. In forward propagation, the model will initialize two embedding matrices for entities and relations respectively, and they will be embedded into embedding matrices. which are the low-dimensional representations of both. After that, the model will concatenate the embeddings $\bar{e_s}$ and $\bar{e_r}$ and feed them into the mainstem and shortcut at the same time.

Mainstem. After $\bar{e_s}$ and $\bar{e_r}$ enter Mainstem, they pass through a 2D convolutional layer with 16 1×1 filters, followed by several convolutional operations with 3×3 filters, normalisation [9] and activation using the *ReLu* [14,16,17] function. The amount of convolution filters of each layer is doubled compared to the previous one. At the end of these operations, the data is convolved using $n \times 1 \times 1$ filters (where n is the final number of channels) to fit the data from the shortcut and leave Mainstem.

Shortcut. The $\bar{e_s}$ and $\bar{e_r}$ from the other branch enter the shortcut, are normalized and dropout, and then fed into a 2D convolutional layer with $32 \times 3 \times 3$ filters for computation. The results of the computation are fed into the Mainstem for summation after being normalized and finally activated by *ReLu* [14].

Score Function. Formally, the score function for ResConvE can be defined as the following equation:

$$\psi = g_a(vec(f_a(cat(\bar{emb}_s, \bar{emb}_r)) \star \omega + \sum f_a(cat(\bar{emb}_s, \bar{emb}_r)) \star \omega')W)emb_o \quad (2)$$

where $\bar{emb}_s$ and $\bar{emb}_r$ represent the embedding tensors of the subjects and relations, emb_s and emb_r, after 2D reshapings, while $\star$ denotes the convolution operation. emb_o denotes the entity embedding matrices, W is the matrix of weights to be learned. f_a and g_a represent *ReLU* and *Logistic Sigmoid* respectively.

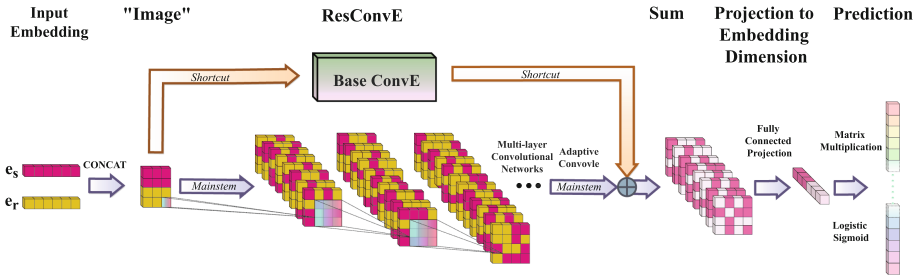

Fig. 2. After feeding the embedding of entities and relations into ResConvE, they will be reshaped and concatenated. Mainstem will then perform three convolutional calculations, while Shortcut will perform a ConvE-like convolutional operation and an adaptive convolutional calculation. The Tensor is then multiplied with the embedding space matrix and a logistic sigmoid is applied to generate a prediction.

5 Experiments

5.1 Knowledge Graph Datasets

FB15k: The FB15k [3] dataset consists of textual mentions of knowledge base relationship triples and Freebase entity pairs. There are 592,213 triples. The number of entities and relationships is 14,951 and 1,345 respectively. FB15K-237 [24] removed the inverse relations.

WN18: The WN18 [13] dataset, with 18 relationships and 40,943 entities, tends to obey a strict grading structure. WN18RR [4] is a new version of WN18 that has emerged from extensive research.

YAGO3-10: YAGO [21] is a KG composed of common knowledge facts extracted from Wikipedia to enhance WordNet. YAGO3-10 has 123,182 entities and 37 relations.

SKG-Course: The SKG-course dataset is derived from the knowledge graph of the *SCHOLAT course platform* (https://www.scholat.com/home.html?type=5), which has a total of 22,176 entities including users, courses, and classes, with the corresponding 4 relationships. Several Baseline models were replicated and trained and tested on this dataset.

5.2 Evaluation Protocol

We test performance through a widely used evaluation process [3,4,26]. We remove the subject or object from the complete triplets in the test set to create corrupted triplets of the form $(subject, relation, ?)$ or $(?, relation, object)$. The

Hits@k and the Mean Rank (MR), and the Mean Reciprocal Rank (MRR) are calculated for evaluation.

Hits@k represents the percentage of entities with the correct subject or object in the top k of all predictions. The MR represents the average ranking of the correct subject or object in the prediction. The MRR represents the mean of the inverse of the ranking of the correct results for multiple predictions, which can normally be calculated by the following formula:

$$MRR = \frac{1}{|pred|} \sum_{i=1}^{|pred|} \frac{1}{rank_i} \tag{3}$$

where $|pred|$ represents the total amount of predictions and $rank_i$ is the ranking of the correct object in the ith prediction.

5.3 Experimental Setup

To verify that ResConvE can extract features better when Mainstem is deepened, we set up three control groups. After the data had passed the first convolutional layer of Mainstem, we set up $\{1, 2, 3, 4\}$ convolutional layers for training respectively, where each convolutional layer had twice the number of filters as the previous one. The experiments were tested on the FB15k and SKG-course datasets respectively.

We tuned the hyperparameters by the performance of MRR. The dropout [9] of the embedding layers, 2D CNN layers and projection layer are set as $\{0.1, 0.2\}$, $\{0.2, 0.3\}$ and $\{0.2, 0.3\}$ respectively. The size of the embedding matrix and batch are set as $\{100 \times 100, 200 \times 200\}$ and $\{128, 256, 512\}$ respectively. We set the learning rate as $\{0.1, 0.001, 0.002, 0.003\}$. The label smoothing [23] coefficient is set as $\{0.1, 0.2\}$.

6 Result

6.1 Comparison of Performance

Comparison with Existing Methods. Besides the benchmarks dataset, we compared the performance of ResConvE with several existing methods on the SKG-course dataset to test generalization capabilities. We replicated several basic models to perform link prediction on the SKG-course to obtain training scores. Table 2, Table 3 and Table 4 summary the performance of ResConvE on the standard dataset and the SKG-course respectively. We find that ResConvE outperforms some metrics on FB15k, WIN18 dataset, and YAGO3-10, while all metrics are better on SKG-course. The results of ResConvE's link prediction on SKG-course are higher in MR metrics compared to ConvE and InteractE by 10.12%, 4.35%. On the validation set, ResConvE even outperformed ConvE and InteractE by 13.64% and 16.73% respectively.

Table 2. Performance on dataset FB15k and FB15k-237

	FB15k					FB15k-237				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE [3]	125	–	0.471	–	–	–	0.290	0.470	–	0.290
TransD [10]	91	–	0.773	–	–	–	0.253	0.461	–	0.148
DistMult [29]	97	0.654	0.824	0.733	0.546	254	0.241	0.419	0.263	0.155
ComplEx [25]		0.692	0.840	0.759	0.599	339	0.247	0.428	0.275	0.158
ConvE [4]	51	0.657	0.831	0.723	0.558	244	0.325	0.501	0.356	0.237
InteractE [26]	–	–	–	–	–	172	0.354	0.535	–	0.263
ResConvE	60	**0.708**	**0.851**	**0.803**	**0.762**	272	0.312	0.486	0.341	0.225

Table 3. Performance on dataset WN18 and WN18RR

	WIN18					WIN18RR				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
DistMult [29]	902	0.022	0.936	0.914	0.728	–	0.43	–	–	0.39
ConvE [4]	374	0.943	0.956	0.946	0.935	4187	0.43	0.52	0.44	0.4
HHolE [11]	183	0.939	0.951	0.945	0.931	–	–	–	–	–
LogicENN [15]	357	0.923	0.948	–	–	–	–	–	–	–
InteractE [26]	–	–	–	–	–	5202	0.463	0.528	–	0.43
ResConvE	393	**0.943**	0.954	**0.949**	**0.936**	5006	0.424	0.491	0.435	0.393

Effect of Deepening the Mainstem. We analyzed whether deepening the mainstem would lead to better performance of ResConvE, i.e. by increasing the number of convolutional layers. We analyze this effect on the FB15k, SKG-course dataset respectively, which are shown in Table 5 and Table 6.

After deepening Mainstem, the model with 4 convolutional layers improved significantly for link prediction on dataset FB15k, with 23.08%, 30.27% and 17.54% on MR, MRR and $Hits@10$ respectively.

Meanwhile, We also found that deepening the convolutional layers on the SKG-course dataset resulted in significant improvements for each of the metrics in Table 6. The model with an increased number of convolutional layers of 3 is higher in MR and MRR by 18.739% and 3.101% respectively than the one with

Table 4. Performance on dataset SKG-course and YAGO3-10

	SKG-course					YAGO3-10				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
DisMult [29]	158	0.889	0.923	0.899	0.874	1107	0.500	0.660	0.550	0.410
ComplEx [25]	150	0.971	0.969	0.970	0.968	1127	0.490	0.660	0.540	0.400
ConvE [4]	112	0.931	0.961	0.942	0.915	1676	0.440	0.620	0.490	0.350
InteractE [26]	105	0.970	0.977	0.972	0.966	1671	0.541	0.620	–	0.462
ResConvE	**100**	**0.973**	**0.978**	**0.974**	**0.970**	2157	0.510	**0.664**	**0.558**	0.427

only 1 additional layer. But there is a limit to the optimization of this effect, and we learn that when the number is increased to 4, the increase is not as pronounced. What is clear, however, is that we believe our model is effective for deepening across a range of datasets, i.e. the generalization effect of our model does exist.

Table 5. Effect of deepening the mainstem on dataset FB15k

Number of convolution layers	MR	MRR	Hits@10	Hits@5	Hits@3	Hits@1
1	78	0.544	0.724	0.724	0.602	0.443
2	60	0.657	0.816	0.762	0.716	0.567
3	63	0.651	0.810	0.755	0.708	0.561
4	60	**0.708**	**0.851**	**0.803**	**0.762**	**0.626**

Table 6. Effect of deepening the mainstem on dataset SKG-course

Number of convolution layers	MR	MRR	Hits@10	Hits@5	Hits@3	Hits@1
1	123	0.944	0.969	0.963	0.957	0.929
2	113	0.947	0.969	0.964	0.958	0.937
3	**100**	**0.973**	0.978	**0.976**	0.973	**0.971**
4	101	0.972	**0.978**	0.975	**0.974**	0.968

Analysis of Experimental Results. Deepening the convolutional neural network to do the link prediction task seems effective. We have analyzed the reason for this: An embedding operation for low-dimensional representation usually means information compression [22]. ResConvE uses multiple CNN layers for modeling, which makes sure of fully learning and extracting the features of the entities and relationships while the Shortcut mechanism introduced by ResConvE ensures that the model is deepened without making the original performance worse.

7 Conclusion and Future Work

In this paper, a new method for KGs embedding, ResConvE, is proposed, which has a better capability of extracting the features of the KG by deepening the neural network, improving the model depth from the same type of ConvE and InteractE. At the same time, ResConvE borrowed the idea of skip connection on residual networks to alleviate the possible gradient disappearance and gradient explosion when the model is deepened and set up Mainstem and shortcut

pathways on the model in learning data respectively. Through extensive experiments, we find that the idea of deepening neural networks has a role in optimizing performance. Moreover, we introduced the SKG-course dataset to demonstrate that this effect is not useful only on specific datasets, but has some generalization ability. We believe that ResConvE can be improved from more angles in the future. Although in this paper we have only made improvements in the depth of the neural network, we believe that improvements could perhaps be made in the width as well. If the model is constructed from CNNs, we believe that there are a large number of tricks in the field of computer vision that can be borrowed into the field of link prediction. We will look at this aspect.

References

1. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
2. Bollacker, K., Evans, C., Paritosh, P.K., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD Conference (2008)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), pp. 1–9 (2013)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
5. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 601–610 (2014)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
7. He, K., Sun, J.: Convolutional neural networks at constrained time cost. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5353–5360 (2015)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456. PMLR (2015)
10. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 687–696 (2015)
11. Lalisse, M., Smolensky, P.: Augmenting compositional models for knowledge base completion using gradient representations. arXiv preprint arXiv:1811.01062 (2018)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)

13. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: ICML (2010)
15. Nayyeri, M., Xu, C., Lehmann, J., Yazdi, H.S.: LogicENN: a neural based knowledge graphs embedding model with logical rules. arXiv preprint arXiv:1908.07141 (2019)
16. Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S.: Activation functions: comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378 (2018)
17. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks. arXiv preprint arXiv:1505.00387 (2015)
20. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. arXiv preprint arXiv:1507.06228 (2015)
21. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706 (2007)
22. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
24. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, pp. 57–66 (2015)
25. Trouillon, T., Dance, C.R., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Knowledge graph completion via complex tensor factorization. arXiv preprint arXiv:1702.06879 (2017)
26. Vashishth, S., Sanyal, S., Nitin, V., Agrawal, N., Talukdar, P.: InteractE: improving convolution-based knowledge graph embeddings by increasing feature interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3009–3016 (2020)
27. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
28. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
29. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
30. Zou, X.: A survey on application of knowledge graph. J. Phys. Conf. Ser. **1487**, 012016 (2020)