Attention-based Direct Interaction Model for Knowledge Graph Embedding

Bo Zhou[1,2], Yubo Chen[1], Kang Liu[1,2], and Jun Zhao[1,2]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
{bo.zhou,yubo.chen,kliu,jzhao}@nlpr.ia.ac.cn

Abstract. Knowledge graph embedding aims at learning low-dimensional representations for entities and relations in knowledge graph. Previous knowledge graph embedding methods usually assign a score to each triple in order to measure the plausibility of it. Despite of the effectiveness of these models, they ignore the fine-grained(matching signals between entities and relations) clues since their scores are mainly obtained by manipulating the triple as a whole. To address this problem, we instead propose a model which firstly produce diverse features of entity and relation by multi-head attention and then introduce the interaction mechanism to incorporate matching signals between entities and relations. Experiments show that our model achieves better link prediction performance than multiple strong baselines on two benchmark datasets WN18RR and FB15k-237.

Keywords: Knowledge graph embedding · Link prediction · Attention-based · Direct interaction.

1 Introduction

Knowledge graphs can be regarded as large knowledge bases(KBs) which consist of structured triples in the form (entity, relation, entity). There are many KBs, such as DBpedia [15], YAGO [23] and Freebase [2] which can offer great help in many natural language processing applications such as relation extraction [10, 28, 20], question answering [5, 3, 8] and machine reading comprehension [30]. However, these KBs are far from complete, that is to say, many valid facts aren't contained in the KBs. Therefore, many researches have been focused on the task *knowledge base completion* which aims to predict the tail entity when given the head entity and relation, or vice versa.

In order to conduct the *knowledge base completion* task, different models have been proposed in recent years. Roughly, these can be divided into two categories [26], one is *Translational Distance Models* and they measure the plausibility of a fact as the distance between the two entities, usually after a translation carried out by the relation. The other is *Semantic Matching Models* and they measure plausibility of facts by matching latent semantics of entities and relations embodied in their vector space representations.

These models can learn good representation for entities and relations in KBs and perform well in knowledge base completion task. Despite of the effectiveness, they ignore the fine-grained clues(matching signals between entities and relations) since their scores are mainly obtained by manipulating the triple as a whole. Take the model ConvE [6] for example, for each fact triple (h, r, t), the concatenation of embeddings of h and r is input to CNN to obtain different feature maps, which will be concatenated and projected to a vetor of the same size as entity embedding of t. Then logit for each entity will be obtained by dot product between the projected vector and the corresponding entity embedding. The ConvE simply extracts a vector from embeddings of h and r and the dot product can only capture linear relationships.

To address this problem, we instead propose a model which incorporates multi-head attention and interaction mechanisms. First, diverse features of head entity and relation are produced by multi-head attention. Then we introduce the interaction mechanism to calculate matching scores between these features and entities and these interaction scores will be further processed to abtain logit for each entity. Besides, we include a nonlinear transformation to better evaluate the dot product.

In summary, our contributions in this paper are as follows:

- We propose a model which leverages the interaction mechanism to directly calculate the interaction signals between different feature vectors and embeddings of entities.
- We utilize multi-head attention to produce diverse features and nonlinear transformation to better evaluate the dot product.
- We evaluate our model on two benchmark datasets and our model achieves better link prediction performance than multiple strong baselines.

2 Related Work

Translational Distance Models use additive functions over embeddings to obtain a score. TransE [4] is the first model to introduce translation-based embedding, which represents both entities and relations as real vectors of same length. It assumes $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$ and minimizes $f_r(h, t) = \|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_{1/2}$. TransH [27] introduces relation-specific hyperplanes in order to better model the 1-to-N, N-to-1 and N-to-N relations which can't be well dealt with in TransE. TransR [16] is similar to TransH, but it introduces relation-specific spaces instead of hyperplanes. In order to reduce the number of parameters, TransD [11] decomposes the relation-specific matrix in TransR into product of two vectors. TranSparse [12] also reduces the parameters in TransR by utilizing sparse relation-specific matrix. There are also some works considering the uncertainty in knowledge graph and modeling entities and relations as random vectors [9, 29].

Semantic Matching Models use product-based functions over embeddings to obtain a score. RESCAL [19] represents each entity as a vector and each ralation as a matrix. It defines the score of a triple by $f_r(h, t) = \boldsymbol{h}^\top \boldsymbol{M}_r \boldsymbol{t}$ rather

than the translational distance in TransE. In order to reduce the number of parameters, DistMult [31] replace the relation matrix in RESCAL with a diagonal matrix. ComplEx [24] extends DistMult by introducing complex-valued embeddings which can better model asymmetric relations. There are also some works using neural netwok to calculate the score for each triple. MLP [7] first maps embeddings of entities and relations into hidden representations which will be added up, then the score is obtained by dot product. ConvKB [18] uses CNN to produce feature maps and then calculated the score by dot product.

3 The Proposed Model

A knowledge graph $\mathcal{G}$ can be seen as a set which contains valid triples (head entity, relation, tail entity) denoted as (h, r, t) such that $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$ where $\mathcal{E}$ is a set of entities and $\mathcal{R}$ is a set of relations. Each embedding model aims to define a score for a triple such that valid triples get higher scores than invalid ones. Table 1 gives score functions of some previous SOTA models.

Table 1. The score functions in previous SOTA models. $\langle \boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t \rangle = \sum_i \boldsymbol{v}_{h_i} \boldsymbol{v}_{r_i} \boldsymbol{v}_{t_i}$ denotes a tri-linear dot product. *Re* is an operation to take the real part of a complex number.

Model	The score function $f(h, r, t)$
TransE	$-\left\|\boldsymbol{v}_h + \boldsymbol{v}_r - \boldsymbol{v}_t\right\|_{1/2}$
ComplEx	$Re\left(\langle \boldsymbol{v}_h, \boldsymbol{v}_r, \overline{\boldsymbol{v}}_t \rangle\right)$
DistMult	$\langle \boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t \rangle$
MLP	$\boldsymbol{w}^\top \tanh\left(\boldsymbol{M}^1\boldsymbol{h} + \boldsymbol{M}^2\boldsymbol{r} + \boldsymbol{M}^3\boldsymbol{t}\right)$
ConvE	$\mathbf{f}(\mathrm{vec}(f([\overline{\mathbf{r}}, \overline{\mathbf{h}}] * \boldsymbol{\Omega}))\mathbf{W})\mathbf{t}$
RotatE	$-\left\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\right\|^2$

We have a lookup table for entities $\boldsymbol{E} \in \mathbb{R}^{|\mathcal{E}| \times d}$ named entity table and a lookup table for relations $\boldsymbol{E} \in \mathbb{R}^{|\mathcal{R}| \times d}$ named relation table, where d denotes the embbeding size.

Given a triple (h, r, t), we first use h and r to look up the entity table and relation table to get their corresponding embeddings denoted as a matrix $\boldsymbol{L} = [\boldsymbol{v}_h, \boldsymbol{v}_r] \in \mathbb{R}^{d \times 2}$. Then, we use the multi-head attention [25]. Given a matrix of n query vectors $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$, keys $\boldsymbol{K} \in \mathbb{R}^{n \times d}$ and values $\boldsymbol{V} \in \mathbb{R}^{n \times d}$, the multi-head attention computes the attention scores based on the following equation:

$$\begin{cases} \boldsymbol{F} = \mathrm{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{Concat}\left(\mathrm{head}_1, \ldots, \mathrm{head}_\mathrm{h}\right) \boldsymbol{W}^O \\ \mathrm{head}_\mathrm{i} = \mathrm{Att}\left(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V\right) = \boldsymbol{f}\left(\frac{(\boldsymbol{Q}\boldsymbol{W}_i^Q)(\boldsymbol{K}\boldsymbol{W}_i^K)^T}{\sqrt{d}}\right)(\boldsymbol{V}\boldsymbol{W}_i^V) \end{cases} \tag{1}$$

here $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are the same matrix and equal to $\boldsymbol{L}^T$, f is softmax and $\frac{1}{\sqrt{d}}$ is the scaling factor. The output of attention layer is a matrix denoted as $\boldsymbol{F} \in \mathbb{R}^{2 \times d}$. We use $\frac{k}{2}$ multi-head attention layers, so after concatenating outputs of these

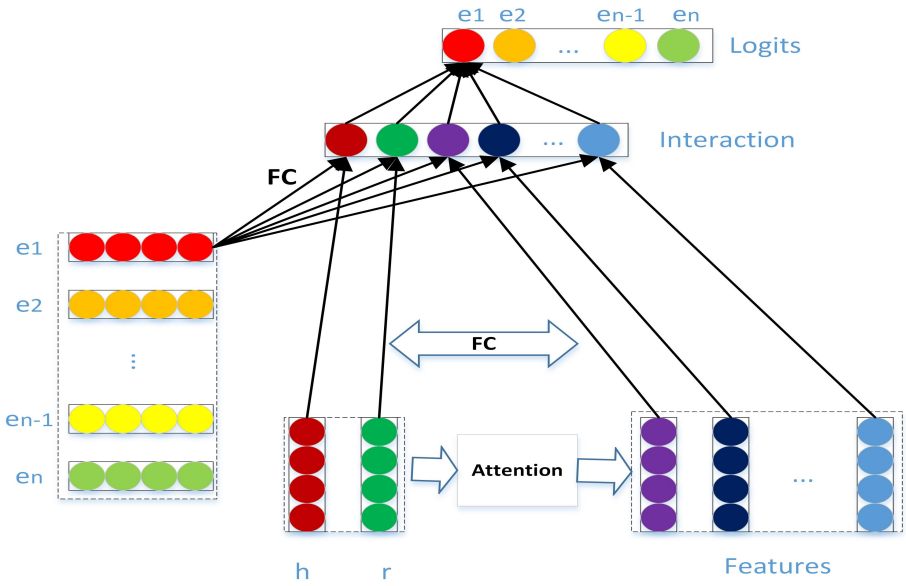

Fig. 1. Illustration of the model with direct interaction. In the picture FC is abbreviation for fully connected layer and for convenience we only show the interaction for e1.

attention layers with $\boldsymbol{L}$, we obtain input matrix $\boldsymbol{I} \in \mathbb{R}^{d\times(k+2)}$. Alternatively, we also replace the multi-head attention layer with CNN to obtain the input matrix $\boldsymbol{I}$. Specifically, the embedding matrix $\boldsymbol{L}$ is input to a convolutional neural network [14] to get differrent feature maps. Specifically, we have k filters and each is of shape $[f_h, 2]$. Thus each filter will produce a feature map of size $[d + 2n_p - f_h + 1, 1]$, where n_p is the number of zero padding which ensures the height of each feature map equals d. Concatenating these k feature maps with $\boldsymbol{L}$, we obtain input matrix $\boldsymbol{I} \in \mathbb{R}^{d\times(k+2)}$.

After obtaining the input matrix $\boldsymbol{I}$ by multi-head attention or CNN, it's input to a fully connected layer(FC):

$$\bar{\boldsymbol{I}} = f\left(\boldsymbol{I}^T\boldsymbol{W}_i + \boldsymbol{b}_i\right) \tag{2}$$

here $\boldsymbol{W}_i \in \mathbb{R}^{d\times d}$ is a weight matrix, $\boldsymbol{b}_i \in \mathbb{R}^d$ is a weight vector and f is an activation function.

The entity table $\boldsymbol{E}$ is also input to an FC:

$$\bar{\boldsymbol{E}} = f\left(\boldsymbol{E}\boldsymbol{W}_e + \boldsymbol{b}_e\right) \tag{3}$$

here $\boldsymbol{W}_e \in \mathbb{R}^{d\times d}$ is a weight matrix, $\boldsymbol{b}_e \in \mathbb{R}^d$ is a weight vector and f is an activation function.

With $\bar{\boldsymbol{E}}$ and $\bar{\boldsymbol{I}}$, we then adopt vector dot product as the interaction function. It's easy to see that the (i, j) entry of $\boldsymbol{D}$ is the dot product of ith row of $\bar{\boldsymbol{E}}$ and jth column of $\bar{\boldsymbol{I}}$, which represents the matching score between ith entity and jth feature of $\bar{\boldsymbol{I}}$:

$$\boldsymbol{D} = \bar{\boldsymbol{E}}\bar{\boldsymbol{I}} \tag{4}$$

here $\boldsymbol{D}$ is called interaction matrix and $\boldsymbol{D} \in \mathbb{R}^{|\mathcal{E}|\times(k+2)}$.

In order to get logits for all entities, we aggregate each row of interaction matrix $\boldsymbol{D}$. First, $\boldsymbol{D}$ is input to an FC:

$$\bar{\boldsymbol{D}} = f\left(\boldsymbol{D}\boldsymbol{W}_d + \boldsymbol{b}_d\right) \tag{5}$$

here $\boldsymbol{W}_d \in \mathbb{R}^{(k+2)\times(k+2)}$ is a weight matrix, $\boldsymbol{b}_e \in \mathbb{R}^{(k+2)}$ is a weight vector and f is an activation function. As for aggregation, there are different ways to implement it such as summation, average, attention and even neural networks. Here we adopt dot product for simplicity and more complicated aggregation strategies are left for future work:

$$\boldsymbol{l} = \bar{\boldsymbol{D}}\boldsymbol{w}_a \tag{6}$$

where $\boldsymbol{w}_a \in \mathbb{R}^{(k+2)\times 1}$ is trainable parameter and $\boldsymbol{l} \in \mathbb{R}^{|\mathcal{E}|\times 1}$ is logits for all entities. It's easy to see that the ith entry of $\boldsymbol{l}$ is the logit of the ith entity, which is a weighted sum of all the matching scores for the entity although the weight is not normalized. We then normalize the logits $\boldsymbol{l}$ into probability:

$$\boldsymbol{p} = \text{sigmoid}\left(\boldsymbol{l}\right) \tag{7}$$

We choose binary classification loss as our loss function and we optimize our model by minimizing the total loss $\mathcal{L}_{loss}$:

$$\mathcal{L}_{loss} = -\sum_{i=1}^{N}\sum_{j=1}^{|\mathcal{E}|}\left(l_i^j \log\left(p_i^j\right) + \left(1 - l_i^j\right)\log\left(1 - p_i^j\right)\right) \tag{8}$$

4 Experiments

4.1 Datasets

Our experiments are conducted on two benchmark datasets: WN18RR and FB15k-237. WN18RR is a subset of WN18 derived from Wordnet [17] which is a KB whose entities (termed synsets) correspond to senses of words, and relationships between entities define lexical relations. Compared with WN18, WN18RR dose not suffer inverse relation test leakage. FB15k-237 is a subset of FB15k derived from Freebase [2] which is a huge and growing KB for common facts with around 1.9 billion triplets. Unlike FB15k, FB15k-237 does not contain inverse relations. In our experiments, we use the same train/valid/test sets split as in [6]. The detailed statistics of WN18RR and FB15k-237 are showed in Table 3.

4.2 Baselines

We compare our model with several previous methods. Our baselines include DistMult, ComplEx, R-GCN [21], ConvE [6] and A2N[1], some of them are strong baselines. We report the results of DistMult, ComplEx, R-GCN and ConvE from [6]. The result of A2N is reported from [1].

Table 2. Statistics of the experimental datasets.

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	#Triples in train/valid/test		
WN18RR	40943	11	86835	3034	3134
FB15k-237	14541	237	272115	17535	20466

4.3 Evaluation Metrics

The purpose of link prediction or KB completion task [4] is to predict a missing entity given the relation and another entity in the valid triple, i.e, predicting h giving (r, t) or predicting t giving (h, r). Then the results are evaluated based on the rankings of the scores calculated by the score function.

Specifically, for each valid triple, we first replace the head entity or tail entity randomly by all the other entity in $\mathcal{G}$ to produce a set of corrupted triples, i.e.,the negative triple sets for the valid triple. Then the score function will be used to calculate all the scores of corrupted triples together with the valid triple and we rank the results based on the scores. We employ common metrics to evaluate the ranking list: mean rank (MR)(i.e., the mean rank of the correct test triples), mean reciprocal rank (MRR)(i.e., the reciprocal mean rank of the correct test triples), Hits@10(i.e., the proportion of the correct test triples ranked in top 10 predictions), Hits@3 and Hits@1.

As pointed out in [4], the above corrupted triple set for each test triple may contain some valid triples in $\mathcal{G}$ and these valid triples may be ranked above the test triple. To avoid this, we follow [4] to remove from the corrupted triple set all the triples that appear in $\mathcal{G}$. The former is called Raw setting and the latter is called Filter setting. We then evaluate results with MR, MRR, Hit@10, Hit@3 and Hit@1 on the new ranking list.

4.4 Training Protocol

We use the common Bernoulli trick [27, 16] to generate the head or tail entities when producing negative triples, i.e.,with probabily p the head entity of a test triple is replaced and with probabily $1-p$ the tail entity is replaced. We calculate MR, MRR, Hit@1, Hit@3 and Hit@10 in Filter setting.

In our experiment, the dimension d of entities and relations are all 200. We initialize all the lookup tables by uniform distribution between $\left[-\frac{6.0}{\sqrt{200}}, \frac{6.0}{\sqrt{200}}\right]$.

The number of muti-head attention is 5 and for each attention layer, we set head number to 2. The filters of CNN are initialized by a uniform distribution $\left[-\frac{1}{\sqrt{22}}, \frac{1}{\sqrt{22}}\right]$. The number of filters for CNN is 8 and the filter size is $[2, 11]$. In order to keep the size of feature maps, zero padding of shape $[0, 5]$ is used. We also use Dropout [22] after the CNN and the dropout rate is 0.1. We choose ReLU as the activation function f. The label smoothing (Szegedy et al. 2016) is also used to reduce over-fitting and the ratio is set to 0.1. In order to make the training process better, the trick of learning rate decay is utilized. For both datasets, the batch size is 128. The Adam optimizer [13] is used to train our model. The initial learning rate 0.001 for both datasets. We run our model on both datesets to 1000 epochs and the validation set is used to select the best model in these epochs to do test set evaluation.

4.5 Experimental Results

Results are summarized in Tables 3 and 4. Table 3 compares the performance of our model with results of previous models, from which we can see that our attention-based model outperforms all baselines on FB15k-237 in terms of all the evaluation metrics. This shows the effectiveness of our model. However, compared with attention-based model, the CNN-based model performs worse on FB15k-237.

On WN18RR, our attention-based model achieves best performance as for MR and MRR. In terms of Hit@10, Hit@3 and Hit@1, our attention-based model obtains comparable peformance with ComplEx and A2N. Once again the CNN-based model performs worse compared with attention-based model on WN18RR. We ascribe this to the better capability of capturing the internal structure of entities and relations and muti-head mechanism of our attention-based model.

Table 3. Results on WN18RR and FB15k-237. Best results are in bold.

	WN18RR					**FB15k-237**				
			Hits					Hits		
Model	MR	MRR	@10	@3	@1	MR	MRR	@10	@3	@1
DistMult	5110	.43	.49	.44	.39	254	.241	.419	.263	.155
ComplEx	5261	.44	**.51**	**.46**	.41	339	.247	.428	.275	.158
R-GCN	-	-	-	-	-	-	.248	.417	.258	.153
ConvE	5277	.46	.48	.43	.39	246	.316	.491	.350	.239
A2N	-	.45	**.51**	**.46**	**.42**	-	.317	.486	.348	.232
Ours(CNN)	5423	.43	.44	.41	.38	271	.310	.471	.334	.229
Ours(Attention)	**5005**	**.47**	.48	.44	.39	**240**	**.331**	**.499**	**.357**	**.248**

In Table 4, we study the influence of number of muti-head attention in our attention-based model on FB15k-237. We compare number of 1, 3 and 5 and report the corresponding performance in terms of MRR, Hit@10 and Hit@1. From the table we see that as the number increases the performance tends to get

better in general, which coincides with our intuition. But taking into account model complexity and computing efficiency, we set the number of muti-head attention to 5 in our attention-based model.

Table 4. The influence of number of muti-head attention on FB15k-237.

Num of attention	MRR	Hit@10	Hit@1
1	.320	.487	.239
3	.328	.495	.249
5	.331	.499	.248

5 Conclusion

In this paper, we propose a novel model for the knowledge graph embedding task. The model leverages the interaction mechanism to directly calculate the interaction signals between different feature vectors and embeddings of entities. Meanwhile, the model utilizes multi-head attention to produce diverse features and nonlinear transformation to better evaluate the dot product. Experiments show that our model achieves better performance in link prediction task than multiple strong baselines on two benchmark datasets WN18RR and FB15k-237. In the future, we plan to incorporate more complicated interaction and aggregation mechanisms to better model the knowledge embedding task.

References

1. Bansal, T., Juan, D.C., Ravi, S., McCallum, A.: A2n: Attending to neighbors for knowledge graph inference. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. pp. 4387–4392 (2019)
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250. AcM (2008)
3. Bordes, A., Chopra, S., Weston, J.: Question answering with subgraph embeddings. arXiv preprint arXiv:1406.3676 (2014)
4. 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. pp. 2787–2795 (2013)
5. Bordes, A., Weston, J., Usunier, N.: Open question answering with weakly supervised embedding models. In: Joint European conference on machine learning and knowledge discovery in databases. pp. 165–180. Springer (2014)
6. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

7. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: 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. ACM (2014)
8. Hao, Y., Zhang, Y., Liu, K., He, S., Liu, Z., Wu, H., Zhao, J.: 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). pp. 221–231 (2017)
9. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with gaussian embedding. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 623–632. ACM (2015)
10. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 541–550. Association for Computational Linguistics (2011)
11. 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). vol. 1, pp. 687–696 (2015)
12. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
15. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
16. 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 (2015)
17. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
18. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. arXiv preprint arXiv:1712.02121 (2017)
19. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML. vol. 11, pp. 809–816 (2011)
20. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 74–84 (2013)
21. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. pp. 593–607. Springer (2018)
22. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)

23. 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. ACM (2007)
24. 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 (2016)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
26. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017)
27. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI conference on artificial intelligence (2014)
28. Weston, J., Bordes, A., Yakhnenko, O., Usunier, N.: Connecting language and knowledge bases with embedding models for relation extraction. arXiv preprint arXiv:1307.7973 (2013)
29. Xiao, H., Huang, M., Zhu, X.: Transg: A generative model for knowledge graph embedding. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 2316–2325 (2016)
30. Yang, B., Mitchell, T.: Leveraging knowledge bases in lstms for improving machine reading. arXiv preprint arXiv:1902.09091 (2019)
31. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)