Relation and Fact Type Supervised Knowledge Graph Embedding via Weighted Scores

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 use just one score to measure the plausibility of a fact, which can't fully utilize the latent semantics of entities and relations. Meanwhile, they ignore the type of relations in knowledge graph and don't use fact type explicitly. We instead propose a model to fuse different scores of a fact and utilize relation and fact type information to supervise the training process. Specifically, scores by inner product of a fact and scores by neural network are fused with different weights to measure the plausibility of a fact. For each fact, besides modeling the plausibility, the model learns to classify different relations and differentiate positive facts from negative ones which can be seen as a muti-task method. Experiments show that our model achieves better link prediction performance than multiple strong baselines on two benchmark datasets WN18 and FB15k.

Keywords: Knowledge graph embedding · Relation supervised · Fact type supervised · Weighted scores.

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 [14], YAGO [24] and Freebase [1] which can offer great help in many natural language processing applications such as relation extraction [8, 28, 21], question answering [4, 2, 6] 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. But there are two problems in existing methods, one is that they just use one score to measure the plausibility for each fact triple. Take the classic model TransE [3] for example, for each fact triple (h, r, t), the score is represented as $f_r(h, t) = -\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_2$ by looking up the embedding table and the score is used for subsequent training. We argue that the only one score is too simple to make full use of the latent semantics of entities and relations encoded in the low-dimensional vector representation. The other problem is that they ignore the relation type and don't explicitly use the fact type, most models just minimize a loss like the pairwise ranking loss to encourage positive triples to get high scores and negative ones to get low scores.

To solve the above two problems, we propose a model to fuse different scores of a fact and utilize relation and fact type information. First, scores by inner product of a fact and scores by neural network are fused with different weights. Then for each fact, besides obtaining a score, we use multi-task learning to learn a classifier to differentiate positive facts from negative ones. At last, we add a relation classification loss and a fact type classification loss to the multi-task learning loss to jointly train the model.

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

- We propose a model to fuse different scores of a fact which can fully utilize the latent semantics of entities and relations.
- In order to make better use of relation and fact type information, we use muti-task learning to simultaneously train a relation classifier and fact classifier besides modeling the scores of a triple.
- We evaluate our model on two benchmark datasets and our model achieves better link prediction performance than multiple strong baselines. Besides, we conduct ablation study which shows the effectiveness of both the relation and fact type information and the weighted scores.

2 Related Work

Translational Distance Models use additive functions over embeddings to obtain a score. TransE [3] 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 [15] is similar to TransH, but it introduces relation-specific spaces instead of hyperplanes. In order to reduce the number of parameters, TransD [9] decomposes the relation-specific matrix in TransR into product of two vectors. TranSparse [10] 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 [7, 29].

Semantic Matching Models use product-based functions over embeddings to obtain a score. RESCAL [20] 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 [25] 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 [5] first maps embeddings of entities and relations into hidden representations which will be added up, then the score is obtained by dot product. ConvKB [17] 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 tiples (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	$\boldsymbol{w}^\top \tanh\left(\boldsymbol{M}^1 \boldsymbol{h} + \boldsymbol{M}^2 \boldsymbol{r} + \boldsymbol{M}^3 \boldsymbol{t}\right)$

We have two lookup tables for entities in $\mathbb{R}^{|\mathcal{E}| \times d}$ named head table and tail table respectively and two lookup tables for relations in $\mathbb{R}^{|\mathcal{R}| \times d}$ named relation table and reverse relation table respectively, where d denotes the embbeding size.

Given a triple (h,r,t), we first look up the head table, relation table and tail table to get their corresponding embeddings denoted as a matrix $\boldsymbol{E} = [\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t] \in \mathbb{R}^{d \times 3}$. Then $\boldsymbol{v}_r$ is input to a MLP to conduct relation classification:

$$\boldsymbol{p}_r = \text{softmax}\left(f\left(\boldsymbol{v}_r^T \boldsymbol{W}_r + \boldsymbol{b}_r\right)\right) \tag{1}$$

here $\boldsymbol{W}_r \in \mathbb{R}^{d \times n_r}$ is a weight matrix and $b_r \in \mathbb{R}^{n_r}$ is a weight vector, n_r is the number of relation type and $\boldsymbol{p}_r \in \mathbb{R}^{n_r}$ is a relation probability distribution for $\boldsymbol{v}_r$. Meanwhile, we get score $s_1 = \langle \boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t \rangle$.

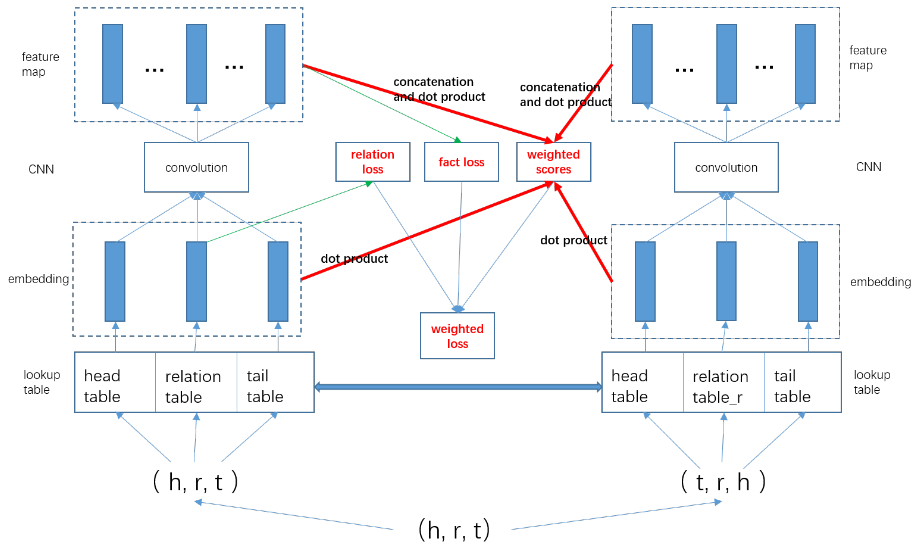

Fig. 1. Illustration of proposed model with relation and fact type supervision via weighted scores.

Then the embedding matrix $\boldsymbol{E}$ is input to a convolutional neural network [13] to get differrent feature maps which will be concatenated to form a vector $\boldsymbol{m}$. Specifically, we have k filters and each is of shape $[f_h, 3]$. Thus each filter will produce a feature map of size $[d - f_h + 1, 1]$. After concatenation of these k feature maps we have $\boldsymbol{m}$ of shape $[k(d - f_h + 1), 1]$ and get score $s_2 = \langle \boldsymbol{m}, \boldsymbol{w} \rangle$ where $\boldsymbol{w}$ is a weight parameter. Meanwhile, another task to classify a fact is conducted with $\boldsymbol{m}$:

$$\boldsymbol{p}_f = \mathrm{softmax}\left(f\left(\boldsymbol{m}^T \boldsymbol{W}_f + \boldsymbol{b}_f\right)\right) \tag{2}$$

here $\boldsymbol{W}_f \in \mathbb{R}^{v_m \times n_f}$ is a weight matrix and $b_f \in \mathbb{R}^{n_f}$ is a weight vector, v_m(i.e., $k(d - f_h + 1)$) is the dimension of vector $\boldsymbol{m}$, n_f is the number of fact type(In our experiment n_f equals 2, namely the positive fact and the negative fact.) and $\boldsymbol{p}_f \in \mathbb{R}^{n_f}$ is a fact probability distribution for $\boldsymbol{m}$.

Similar to [11], for the triple (h, r, t) we also reverse it to get a triple (t, r, h), then we look up the head table, reverse relation table and tail table to get their corresponding embeddings denoted as a matrix $\boldsymbol{E}' = [\boldsymbol{v}'_t, \boldsymbol{v}'_r, \boldsymbol{v}'_h] \in \mathbb{R}^{d \times 3}$ and in the meantime we get score $s'_1 = \langle \boldsymbol{v}'_t, \boldsymbol{v}'_r, \boldsymbol{v}'_h \rangle$. The matrix $\boldsymbol{E}'$ will be input to another CNN of the same structure as described above to get score $s'_2 = \langle \boldsymbol{m}', \boldsymbol{w}' \rangle$. It should be noted that we don't do relation or fact type classification this time. Fig. 1 illustrates the structure of our model.

With score s_1, s_2, s'_1 and s'_2, we define the weighted score function f of our model as follows:

$$f(h, r, t) = \lambda_1 s_1 + \lambda_2 s_2 + \lambda_3 s'_1 + \lambda_4 s'_2 \tag{3}$$

where λ_1, λ_2, λ_3 and λ_4 are weight parameters tuned on validation set.

The loss function $\mathcal{L}_1$ generated by the weighted scores with L_2 regularization on all the embedding vector $\boldsymbol{e}$ of the model is:

$$\mathcal{L}_1 = \sum_{(h,r,t)\in\{\mathcal{G}\cup\mathcal{G}'\}} \log\left(1+\exp\left(l_{(h,r,t)}\cdot f(h,r,t)\right)\right) + \frac{\lambda}{2}\|\boldsymbol{e}\|_2^2 \tag{4}$$

$$l_{(h,r,t)} = \begin{cases} 1 \text{ for } (h,r,t)\in\mathcal{G} \\ -1 \text{ for } (h,r,t)\in\mathcal{G}' \end{cases} \tag{5}$$

where $\mathcal{G}'$ is a set of triples generated by corrupting valid triples in $\mathcal{G}$.

We use cross entorpy loss $\mathcal{L}_2$ as our relation classification loss:

$$\mathcal{L}_2 = -\sum_{i=1}^{N}\sum_{j=1}^{n_r}\left(\boldsymbol{l}_{ri}^j \log\left(\boldsymbol{p}_{ri}^j\right)\right) \tag{6}$$

where $\boldsymbol{l}_r$ is the label vector of relation and N is the number of training instances. Similarly, the fact classification loss $\mathcal{L}_3$ is:

$$\mathcal{L}_3 = -\sum_{i=1}^{N}\sum_{j=1}^{n_f}\left(\boldsymbol{l}_{fi}^j \log\left(\boldsymbol{p}_{fi}^j\right)\right) \tag{7}$$

where $\boldsymbol{l}_f$ is the label vector of fact type and N is the number of training instances.

With $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$, the total weighted loss $\mathcal{L}$ is:

$$\mathcal{L} = \alpha_1\mathcal{L}_1 + \alpha_2\mathcal{L}_2 + \alpha_3\mathcal{L}_3 \tag{8}$$

where α_1, α_2 and α_3 are weight parameters tuned on validation set. We optimize our model by minimizing the total loss $\mathcal{L}$.

4 Experiments

4.1 Datasets

Our experiments are conducted on two benchmark datasets: WN18 and FB15k. WN18 is a subset of Wordnet [16] which is a KB whose entities (termed synsets) correspond to senses of words, and relationships between entities define lexical relations. FB15k is a subset of Freebase [1] which is a huge and growing KB for common facts with around 1.9 billion triplets. In our experiments, we use the same train/valid/test sets split as in [3]. The detailed statistics of WN18 and FB15k are showed in Table 3.

4.2 Baselines

We compare our model with several previous methods. Our baselines include TransE, TransR, STransE [19], NTN [22], DistMult, ComplEx and SimpleE, some of them are strong baselines. We report the results of TransE, STransE, DistMult, and ComplEx from [25]. The results of TransR and NTN are reported from [18], and SimpleE is reported from [11].

Table 2. Statistics of the experimental datasets.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#Triples in train/valid/test		
WN18	40943	18	141442	5000	5000
FB15k	14951	1345	483142	50000	59071

4.3 Evaluation Metrics

The purpose of link prediction or KB completion task [3] 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 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 [3], 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 [3] 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, 15] 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 MRR in both Raw and Filter setting and Hit@1, Hit@3 and Hit@1 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 filters of CNN are initialized by a uniform distribution $\left[-\frac{1}{\sqrt{303}}, \frac{1}{\sqrt{303}}\right]$. The number of filters for CNN is 3 and the filter size is $[3, 101]$. We also use Dropout [23] after the CNN and the dropout rate is 0.5. We choose ReLU as the activation function f. The L_2 regularizer λ is fixed at 0.03. The weight parameters λ_1, λ_2, λ_3, λ_4 α_1, α_2 and α_3 are set to 0.4, 0.2, 0.3, 0.1, 0.6, 0.2 and 0.2 respectively. Besides, we also try attention mechanism to dynamically tune these weight parameters in our experiment. For the WN18 dataset, the batchsize is 1415 and for the FB15k dataset the batchsize is 4832. For both datasets, we sample 1 negative triple for each positive triple. The Adam optimizer [12] is used to train our model with initial learning rate 0.1. 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

Table 3 compares the performance of our model with results of previous models, from which we can see that our model outperforms TransE, TransR, STransE, NTN and DistMult on both datasets for all the valuation metrics. This shows the effectiveness of our model.

Compared with ComplEx and SimplE, on WN18 dataset our model achieves best performance as for MRR and Hit@1. Hit@3 is slightly lower than ComlpEx and we obtain the same Hit@10 as ComlpEx and SimplE. On FB15k dataset, our model obtains lower MRR in the Filter setting and Hit@3 but we achieve best performance as for MRR in the Raw setting, Hit@1 and Hit@10. Besides, our model achieves significant improvement compared with all the baselines as for the MRR in the Raw setting on both datasets. This also shows the effectiveness of our model, in fact ComplEx and SimplE are strong baselines which are previous SOTA models.

Table 3. Results on WN18 and FB15k. Best results are in bold.

	WN18					FB15k				
	MRR		Hit@			MRR		Hit@		
Model	Filter	Raw	1	3	10	Filter	Raw	1	3	10
TransE	0.454	0.335	0.089	0.823	0.934	0.380	0.221	0.231	0.472	0.641
TransR	0.605	0.427	0.335	0.876	0.940	0.346	0.198	0.218	0.404	0.582
STransE	0.657	0.469	-	-	0.934	0.543	0.252	-	-	0.797
NTN	0.530	-	-	-	0.661	0.250	-	-	-	0.414
DistMult	0.822	0.532	0.728	0.914	0.936	0.654	0.242	0.546	0.733	0.824
ComplEx	0.941	0.587	0.936	**0.945**	**0.947**	0.692	0.242	0.599	0.759	0.840
SimplE	0.942	0.588	0.939	0.944	**0.947**	**0.727**	0.239	0.660	**0.773**	0.838
Ours	**0.943**	**0.596**	**0.940**	0.944	**0.947**	0.715	**0.258**	**0.661**	0.770	**0.843**

Ablation Study Table 4 shows the results of our ablation study on WN18, from which we can see that the full model achieves the best performance as for MRR in both Filter and Raw setting. If we ablate Relation classifier or Fact classifier, the performance will degrade. After ablation of both classifier, the model obtains the lowest performance. This demonstrates the effectiveness of both the relation and fact type information for knowledge graph embedding task. Besides, after ablating the Relation classifier the performance degrades more than ablating the Fact classifier, which indicates that relation information is more important than fact type information for knowledge graph embedding task.

Table 4. Ablation study for WN18.

Ablation	MRR(Filter)	MRR(Raw)
Full model	0.943	0.596
Relation classifier	0.940	0.590
Fact classifier	0.940	0.594
Both	0.939	0.580

5 Conclusion

In this paper, we propose a novel model for the knowledge graph embedding task. The model fuses scores by inner product of a fact and scores by neural network with different weights to measure the plausibility of a fact. The model also learns a classifier to classify different relations and differentiate positive facts from negative ones which can be seen as a muti-task method when modeling the plausibility for each fact. Both of these can help to obtain better representations for entities and relations. Experiments show that our model achieves better performance in link prediction task than multiple strong baselines on two benchmark datasets WN18 and FB15k. In the future, we plan to use more complicated fusion strategy and further utilize relation and fact type information to better model the knowledge embedding task.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No.61533018), the Natural Key R&D Program of China (No.2017YFB1002101), the National Natural Science Foundation of China (No.61806201, No.61702512) and the independent research project of National Laboratory of Pattern Recognition. This work was also supported by CCF-Tencent Open Research Fund.

References

1. 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)
2. Bordes, A., Chopra, S., Weston, J.: Question answering with subgraph embeddings. arXiv preprint arXiv:1406.3676 (2014)
3. 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)
4. 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)
5. 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)

6. 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)
7. 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)
8. 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)
9. 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)
10. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
11. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Advances in Neural Information Processing Systems. pp. 4284–4295 (2018)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. 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)
14. 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)
15. 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)
16. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
17. 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)
18. Nguyen, D.Q.: An overview of embedding models of entities and relationships for knowledge base completion. arXiv preprint arXiv:1703.08098 (2017)
19. Nguyen, D.Q., Sirts, K., Qu, L., Johnson, M.: Stranse: a novel embedding model of entities and relationships in knowledge bases. arXiv preprint arXiv:1606.08140 (2016)
20. 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)
21. 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)
22. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems. pp. 926–934 (2013)

23. 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)
24. 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)
25. 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)
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)