Knowledge Graph Embedding Based on Relevance and Inner Sequence of Relations

Jia Peng[1,2], Neng Gao[1], Min Li[1,2(✉)], and Jun Yuan[1,2]

[1] SKLOIS, Institute of Information Engineering, CAS, Beijing, China
{pengjia,gaoneng,minli,yuanjun}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. Knowledge graph Embedding can obtain the low-dimensional dense vectors, which helps to reduce the high dimension and heterogeneity of Knowledge graph (KG), and enhance the application of KG. Many existing methods focus on building complex models, elaborate feature engineering or increasing learning parameters, to improve the performance of embedding. However, these methods rarely capture the influence of intrinsic relevance and inner sequence of the relations in KG simultaneously, while balancing the number of parameters and the complexity of the algorithm. In this paper, we propose a concatenate knowledge graph embedding method based on relevance and inner sequence of relations (KGERSR). In this model, for each $< head, relation, tail >$ triple, we use two partially shared gates for head and tail entities. Then we concatenate these two gates to capture the inner sequence information of the triples. We demonstrate the effectiveness of the proposed KGERSR on standard FB15k-237 and WN18RR datasets, and it gives about 2% relative improvement over the state-of-the-art method in terms of $Hits@1$, and $Hits@10$. Furthermore, KGERSR has fewer parameters than ConmplEX and TransGate. These results indicate that our method could be able to find a better trade-off between complexity and performance.

Keywords: Knowledge graph embedding · Relations relevance · Inner sequence · Cascade model

1 Introduction

KG simulates human understanding of various things and their relations in the real world to construct structured and semantic knowledge representation. Because of the large amount data in real-world KG, an efficient and scalable solution is crucial. KGE is a feature extraction process, mapping a complex network which includes nodes, content, and relations into low-dimensional vector spaces.

Supported by the National Key Research and Development Program of China.

H. Yang et al. (Eds.): ICONIP 2020, CCIS 1332, pp. 78–86, 2020.
https://doi.org/10.1007/978-3-030-63820-7_9

Many KGE methods have been proposed [1,11,13] to learn low-dimensional vectors of entities and relations. In fact, an entity may have multiple aspects which may be related to different relations [8]. Therefore, many independent models [6,8], have been proposed recently and usually outperform dependent models on public datasets. However, current methods always assume the independence between relations and try to learn unique discriminate parameter set for each relation, which leads to a sharp increase in parameters and high time complexity.

Meanwhile, the sequence information in triple should also be taken into account. Although the translation-based model considers the order to some extent by the formula $h + r \approx t$, it is still under exploit to inherent sequence information of the triples.

To optimize embedding performance by considering the relevance and inner sequence, we explore knowledge graphs embedding from two perspectives. On one hand, there is a certain potential connection between the relations, which is neither completely independent nor completely consistent for one entity. On the other hand, the triple should be considered as a sequence, which includes the order information. Based on those ideas, we develop a novel partial layer concatenate mechanism and propose an efficient knowledge graph embedding method based on relevance and inner sequence of relations (KGERSR). It uses a shared concatenate sigmoid layer: one part is two shared filters for discriminating specific relation information of all kinds of relations; the other part is a uniform sequence holder that preserves the inner sequence information of the triple.

We evaluate our method on knowledge graph completion, and the experiments show that our model is comparable to or even outperforms state-of-the-art baselines. The main contributions of this paper are summarized as follows:

- We found that the relations in the heterogeneous KG are not completely independent, while each relation contributes differently to the embedding of one aspect of the entity. Therefore, we propose a scalable and efficient model KGERSR with two gates to discriminate the inherent relevance of relations.
- Besides, the inner sequence of relations needs to be considered in the embedding of the entity. We develop a layer concatenate mechanism to capture the inner sequence information of the triples.
- In order to find a balance between complexity and accuracy, we propose an shared parts of parameter matrix which can preserve correlation and inner sequence information, using three parameter matrices. The complexity is as same order as transE.
- Experiments show that KGERSR delivers some improvements compared to state-of-the-art baselines, and reduces parameters. These results indicate that our method is a good way to further optimize embedding in a real KG.

2 Related Work

Translational Distance Models is one of the representative methods of KG Embedding model. TransE [1] is the earliest translational distance model. It

represents both entities and relations as vectors in the same space. Despite its simplicity and efficiency, TransE has flaws in dealing with 1-to-N, N-to-1, and N-to-N relations [8,14], so that they do not do well in dealing with some complex properties. To overcome the disadvantages of TransE in dealing with complex relations, some method such as transH [14] and transR [8] are proposed, which introduce relation-specific entity embeddings strategy. Those methods need a large-scale of parameters and high time complexity, which prevent them from applying on large-scale KG.

Some works take the relevence of relations into account, assuming the relations fit some sort of random distribution, and modeling them as random variables. KG2E [4] represents entities and relations as random vectors drawn from multivariate Gaussian distributions. TransG [15] also models entities with Gaussian distributions, and it believes that a relation can have multiple semantics, hence it should be represented as a mixture of Gaussian distributions. However, once the entities and relations of the actual KG do not conform to the assumed distribution, the effect of those models will be weakened.

There are many methods based on semantic matching models that also consider the correlation between relations to reduce learning parameters. DistMult [16] introduces a vector $\mathbf{r}$ and requires $\mathbf{M}_r = diag(\mathbf{r})$. The scoring function is hence defined as $f_r(h,t) = \mathbf{h}^\top diag(\mathbf{r})\mathbf{t}$. This score captures pairwise interactions between only the components of h and t along the same dimension, and reduces the number of parameters to $\mathcal{O}(d)$ per relation. However, this oversimplified model can only deal with symmetric relations which is clearly not powerful enough for general KG. ComplEx [13] extends DistMult by introducing complex-valued embeddings so as to better model asymmetric relations.

In the KG, the sequence of relations can also reflect the semantic relation between entities. Lin proposed a representation learning method Path-based TransE (PTransE)[9]. Given a path p linking two entities h and t, p can be calculated using the addition, multiplication, or RNN of all r_i on the path. Guu et al. [3] proposed a similar framework, the idea of which is to build triples using entity pairs connected not only with relations but also with relation paths. Those models considering relational paths can greatly improve the discrimination of knowledge representation learning and improve performance on tasks such as knowledge graph completion. However, they both had to make approximations by sampling or pruning to deal with the huge number of paths.

3 Our Model

3.1 Motivation

Relations of KG are relevant for each entity. <Arnold Schwarzenegger, isGovernor, California> and <Arnold Schwarzenegger, isMemberOf, Republican> jointly infer to <Arnold Schwarzenegger, is, Politician>. Therefore, we should not completely separate the relations, or embedding together indiscriminately.

By building the learning network, the model can automatically learn the intrinsic relevance between the relations, and let the related work together to express the characteristics of one aspect of the entity.

Additionally, KG is a directed graph, which head entity and tail entity have inner sequence connected by relation. The entities connected by the order relations will affect each other. Consequently, as mentioned before, path in a triple can also reflect the semantic relation between entities. A model capable of capturing sequence information should be proposed. Although the translation-based models handle the path information, which have preserved a certain information of the sequence, they still under exploit to inherent sequence information of the triples.

We should consider retaining the relevance of the relations while retaining the sequence information. So, we combine the ideas of LSTM [5] and RNN [12]. For the relevance of relations, we hope that related relations work and irrelevant relations are ignored, so we design two shared gates for head and tail entities embedding with relations, which draw on the core idea of LSTM. For sequence of relations, we consider the triples as a sequence combining by $[h, r]$ and $[r, t]$. So we develop a recurrent discriminate mechanism to retain the sequence information which draw on the core idea of RNN.

3.2 KGE with Relevance and Sequence of Relations

The framework of KGERSR is shown as Fig. 1 and the detailed descriptions are as follow:

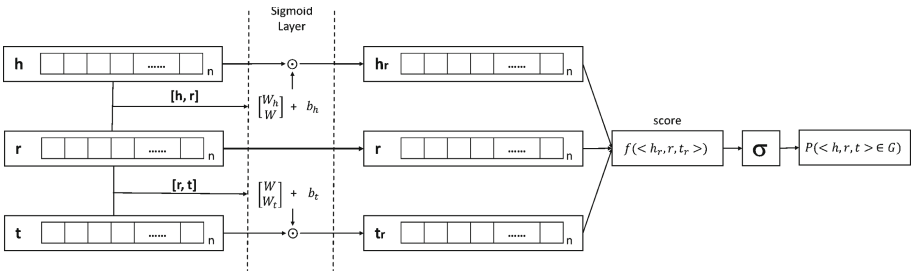


Fig. 1. The KGERSR architecture.

- We map every entity and relation into continues vector with same dimension R^m. Then we get original h, r, t embedding vectors.
- We input both entity embedding and relation embedding into the concatenate sigmoid layer consisted by parameter matrix, which determined by entity and relation together.
- Then we set two shared gates, σ_h and σ_t for head and tail entities respectively. Those two gates partially shared one recurrent parameter matrix W between $[h, r]$ and $[r, t]$.

- We realize non-linear and adaptive relation-specific discrimination through multiplying the different parts of concatenate layer output and entity embedding element-wise.
- We capture the inherent sequence property through multiplying the shared part of concatenate layer output and entity embedding element-wise.
- Last, we build a scoring function useing discriminated information of heads and tails. Based on the score, we can determine whether the triple is valid.

Each triple is composed of two entities and their relation. We define $h, t, r \in R^m$ as their embeddings respectively. The parameters of concatenate layer are denoted as W_h, W_t, $W \in R^{m \times m}$ and b_h, $b_t \in R^m$. W_h and W_t record the relevance between the relations, respectively. W is first affected by $[h, r]$ and then by $[r, t]$, so that W can record the inner sequence of the whole triple. The discriminated vectors of entities are defined as

$$h_r = [h] \odot \sigma([W_h, W] \cdot \begin{bmatrix} h \\ r \end{bmatrix} + [b_h]) \tag{1}$$

$$t_r = [t] \odot \sigma([W, W_t] \cdot \begin{bmatrix} r \\ t \end{bmatrix} + [b_t]) \tag{2}$$

The sigmoid function $\sigma(\bullet)$ is applied in a element-wise manner. [,] means the concatenate operation. $\odot$ means the element-wise product.

In practice, we enforce constraints on the norms of the embeddings. That is to say, $\forall h, t, r$, we have $||h||_2 = 1$, $||r||_2 = 1$, $||t||_2 = 1$, $||h_r||_2 = 1$ and $||t_r||_2 = 1$. The output of concatenate sigmoid layers describes how much relation-specific and sequence information should be maintained.

Then, we define the scoring function f as Eq. 3. The score is expected to be higher for a valid triple and lower for an invalid triple.

$$f(< h, r, t >) = \sum_{k=1}^{m} h_{rk} r_k t_{rk} \tag{3}$$

The log-odd of the probability that G holds the triple is true is:

$$P(< h, r, t > \in G) = \sigma(f(< h, r, t >)) \tag{4}$$

3.3 Training

We use the Adam optimizer [7] to minimize the loss function [13] with L_2 regularization on weight matrix W_h, W_t and W of concatenate layer.

$$\begin{aligned} L = & \sum_{<h,r,t> \in \{G \cup G'\}} log(1 + exp(-Y_{hrt} f(< h, r, t >))) \\ & + \frac{\lambda}{2}(||W_h||_2^2 + ||W_t||_2^2 + ||W||_2^2) \end{aligned} \tag{5}$$

where $Y_{hrt} = 1$ if $< h, r, t > \in G$, and $Y_{hrt} = -1$ otherwise. G' is a collection of invalid triples generated by replacing entities or relations in training triples randomly. We use $\eta = \frac{|G'|}{|G|}$, which is an important hyperparameter, to represent negative samples per training sample. G' is defined as

$$G' = \{< h', r, t > | h' \in E\} \cup \{< h, r, t' > | t' \in E\} \cup \{< h, r', t > | r' \in R\} \tag{6}$$

In practice, we initialize the embeddings, weight matrices and weight vectors of gates through sampling from a truncated standard normal distribution. We use Adam optimizer with a constant range of learning rates for each epoch to carried out the training process which is stopped based on model's performance.

3.4 Complexity Analysis

The parameter number of our method is $\mathcal{O}(N_e m + N_r n + 3m^2 + 2m)(m = n)$, and the time complexity is $\mathcal{O}(m^2)$. N_e, N_r represent the number of entities, relations respectively. m, n are the dimension of entity and relation embedding space, respectively. The parameter complexity of KGERSR is almost the same as TransE in an epoch, because $m \ll N_r \ll N_e$ among existing KG. The discriminate parameters brought by the filters can be ignored compare to embedding parameters. Besides, KGERSR do not need any hyper parameter or pre-training to prevent overfitting. This makes KGERSR can be trained easier, so that it can be used to process the real-world KG.

4 Experiments

4.1 Knowledge Graph Completion Results

Knowledge graph completion aims to predict the missing h or t for a relation fact triple $< h, r, t >$. In this task, we need to filter out the correct triples from hybrid triples.

In this task, we use two datasets: WN18RR and FB15K-247, shown as Table 1. Since the datasets are same, we directly copy experimental results of several baselines. For those two datasets, we traverse all the training triples for at most 1000 rounds. We report MRR which is the mean reciprocal rank of all correct entities) and ($Hits@K$) which is the proportion of correct entities ranked in top K as our evaluation metrics.

Table 1. Statistics of datasets

Dataset	#Rel	#Ent	#Train	#Valid	#Test
WN18RR	11	40943	86835	3034	3134
FB15K-237	237	14541	272115	17535	20466

We select the hyper parameters of KGERSR via grid search according to the MRR on the validation set. On WN18RR, the best configurations are: $\lambda = 0.01$, $\alpha = 0.01$, $m = 200$, $B = 120$ and $\eta = 25$. On FB15K-237,the best configurations are: $\lambda = 0.1$, $\alpha = 0.1$, $m = 200$, $B = 500$ and $\eta = 25$. Table 2 shows the evaluation results on knowledge graph completion.

From Table 2, we observe that KGERSR outperforms all of baselines in some metric and achieves comparable results in other metric. On more relational but sparse data set FB15K-237, our method outperforms 2.3% higher at MRR and 1.9% higher at $Hits@1$ than previous best result. Besides, on less relational but dense data set WN18RR, our method achieves comparable results in MRR and $Hits@1$ than previous best results.

Table 2. Experimental results of knowledge graph completion

Model	WN18RR			FB15K-237		
	MRR	Hits@10(%)	Hits@1(%)	MRR	Hits@10(%)	Hits@1(%)
TransE-2013 [1]	0.266	50.1	39.1	0.294	46.5	14.7
DistMult-2015 [16]	0.43	49.0	39	0.241	41.9	15.5
ComplEX-2016 [13]	0.44	51.0	41	0.247	42.8	15.8
ConvE-2018 [2]	0.43	52.0	40.0	0.325	50.1	23.7
ConvKB-2018 [10]	0.248	**52.5**	–	0.396	51.7	–
TransGate-2019 [17]	0.409	51.0	39.1	0.404	**58.1**	25.1
RKGE-2019 [18]	0.44	**53.0**	**41.9**	0.477	55.4	44.2
KGERSR	**0.45**	48.3	40.9	**0.488**	56	**45.02**

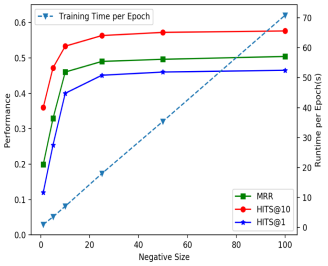
The results indicate that our model is able to achieve more improvements on more relational but sparse graph FB15K-237. The promotion of less relational but dense data set WN18RR is relatively limited. That is to say, our model can perform better in more relational graph. That result shows that the relational relevance part in our model plays a more critical role in KG completion task. The improvement on the indicator $Hits@1$ is obvious, which shows that our algorithm has great ability on precise link prediction. In general, the results demonstrate that our method has a certain generalization. At the same time, the proposed method using concatenate gates which takes into account the relevance and sequence of relations can more fully capture the essential characteristics of entities and relations in KG.

4.2 Parameter Efficiency

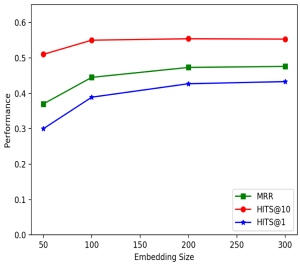
We further investigate the influence of parameters on the performance and the sensitivity to parameters of our model. Experiments below focus on FB15K-237, with the best configurations obtained from the previous experiment. Training was stopped using early stopping based on filtered MRR on the validation set. For the negative samples, embedding size and batch size parameters, we select the parameter values in the corresponding interval one by one, and view the

changes in MRR and the corresponding training time. The result shown in Fig. 2.

- We let negative samples(η) vary in {1, 5, 10, 25, 50, 100}. It can be observed from Fig. 2(a) that generating more negative samples clearly improves the results. When η exceeds 25, the growth rate of indicators such as MRR, $Hits@1$ and $Hits@10$ slows down, and the results are basically flat, while the corresponding training time still increases linearly. So considering the training time, we choose $\eta = 25$ as the optimal parameter.
- We let embedding size(m) vary in {50, 100, 200, 300}. It can be observed from Fig. 2(b) that the embedding size improves the model performance from 50 to 200 interval, but the performance of $Hits@10$ decreases from 55.4% to 55.3% when $m = 300$. Simultaneously, the training time of the model also increases accordingly. In other words, blindly increasing the size of the matrix does not guarantee the effectiveness of the model. The reason may be that increasing the embedding size make some triples trained less inadequately, which makes it impossible to learn accurate embedding results. So considering the model performance, we choose $m = 200$ as the optimal parameter.



(a) Performance and Epoch Time Effected by Negative Size



(b) Performance Effected by Negative Size

Fig. 2. Performance of the different parameter size

5 Conclusion and Future Work

In this paper, in order to find a balance between complexity and accuracy, we propose an shared parts of parameter matrix which can preserve correlation and sequence information, using three parameter matrices. Experiments show that KGERSR outperforms of state-of-the-art baselines in some indicators and achieves comparable results in other indicators. These results indicate that our method is a good way to further optimize embedding in the real KG.

In the future, we will conduct further study from the following aspects: (1) Add information such as text and attributes to further increase the accuracy of knowledge embedding. (2) We will use some sophisticated methods like RNNs to further optimize KGERSR methods. (3) The connection between triples is closer to the graph model.

References

1. Antoine, B., Nicolas, U., Alberto, G.D., Jason, W., Oksana, Y.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), pp. 1–9 (2013)
2. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. Proc. AAAI, 1811–1818 (2017)
3. Guu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space. In: Proceeding of Empirical Methods Natural Language Process, pp. 318–327 (2015)
4. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with gaussian embedding. In: The 24th ACM International Conference on Information and Knowledge Management (CIKM), pp. 623–632, October 2015
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
6. Ji, G., Kang, L., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: AAAI Conference on Artificial Intelligence (2016)
7. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2015)
8. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. Proc. AAAI, 2181–2187, January 2015
9. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. Comput. Sci. (2015)
10. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.Q.: A novel embedding model for knowledge base completion based on convolutional neural network. Proc. NAACL (2018)
11. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Lake Tahoe (2013)
12. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentimenttreebank. In: Proceedings of EMNLP, pp. 1631–1642 (2013)
13. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of ICML, pp. 2071–2080 (2016)
14. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI Conference on Artificial Intelligence (2014)
15. 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, vol. 1: Long Papers, pp. 2316–2325 (2016)
16. Yang, B., Yih, S.W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR), May 2015
17. Yuan, J., Gao, N., Xiang, J.: Transgate: Knowledge graph embedding with shared gate structure. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3100–3107, July 2019
18. Yuan, J., Gao, N., Xiang, J., Tu, C., Ge, J.: Knowledge graph embedding with order information of triplets. In: Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., Huang, S.-J. (eds.) PAKDD 2019. LNCS (LNAI), vol. 11441, pp. 476–488. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16142-2_37