IJLAI Transactions on Science and Engineering (ISSN: 3006-3418) 2024, vol. 2, no. 3, pp: 9–18

Coupling Relation Strength with Graph Convolutional Networks for Knowledge Graph Completion

Mingshao Xu[1], Hang Li[1*], and Zhi Hu[1]

Software College, Shenyang Normal University
No.253, HuangHe North Street, Huanggu District, Shenyang, 110034 China
Corresponding author: Hang Li
Xumingshao6@163.com;lihangsoft@163.com;hu-zhi521@163.com
Received June. 26, 2024; Revised and Accepted July. 2, 2024

Abstract. In the link prediction task of knowledge graph completion, Graph Neural Network (GNN)-based knowledge graph completion models have been shown by previous studies to produce large improvements in prediction results. However, many of the previous efforts were limited to aggregating the information given by neighboring nodes and did not take advantage of the information provided by the edges represented by relations. To address the problem, Coupling Relation Strength with Graph Convolutional Networks (RS-GCN) is proposed, which is a model with an encoder-decoder framework to realize the embedding of entities and relations in the vector space. On the encoder side, RS-GCN captures graph structure and neighborhood information while aggregating the information given by neighboring nodes. On the decoder side, RotatE is utilized to model and infer various relational patterns. The models are evaluated on standard FB15k, WN18, FB15k-237 and WN18RR datasets, and the experiments show that RS-GCN achieves better results than the current state-of-the-art classical models on the above knowledge graph datasets.

Keywords: Knowledge Graph Completion, Graph Convolutional Networks, Relation strength, Link prediction.

1. Introduction

Knowledge Graphs (KGs) [1] such as DBpedia [2] and Freebase [3] consist of nodes (entities) and edges (relations between entities). In semantic Web contexts, statements in a knowledge graph can be represented as a triple (h, r, t), where h represents he head entity, t represents the tail entity, and r represents the relation between the head entity and the tail entity. Knowledge graphs have attracted increasing attention in several domains, such as question and answer [4,5], recommender systems [6] and search engines. However, knowledge graphs remain incomplete as the real world continues to change, which motivates Knowledge Graph Completion (KGC) tasks to predict missing facts and improve the quality of knowledge graphs. Link prediction has become a major focus of Statistical Relational Learning (SRL) in these inference tasks.

While translation-based models are intuitive and easy to understand by capturing the semantics of relations through simple translation vector representations, they cannot capture complex relations and perform poorly in complex knowledge graphs. In contrast, the model based on coded semantic matching is relatively flexible in modeling interactions between entities and relations, but it is less interpretable. In addition, CNN-based models are able to effectively capture the local patterns of entities and relations through convolutional layers, but some global information is lost. Graph neural networks have been shown to be very effective in link prediction tasks in recent experiments. However, most of the previous efforts were limited to aggregating the information given by neighboring nodes and did not take advantage of the information provided by the edges represented by relations. Specifically, most existing GNN-based models do not take advantage of the strengths available in different relations or the same relations between different entities. As in Fig.1, the entity Robert Downey Jr. has two different relations ”Acted in” and ”Collaborated with”, and the different relations have different relation strengths, as noted in previous results such as R-GCN. However, in a relation such as ”Acted in”, where the entity Robert Downey Jr. acted in Iron Man as the sole star, in The Avengers as the star, and in Spider-Man: Homecoming as a special appearance, it is clear that the relation has different strengths of relationship between the different entities;Under the relation ”Collaborated with”, the entity Robert Downey Jr. has a collaborative relationship with the director Joss Whedon, as well as having that collaborative relationship with the actor Chris Evans, which again suggests that the strength of the relationship varies between different entities facing the same relation.

This paper proposes a new graph neural network named RS-GCN to solve the above problem. The key idea is to give different strengths to the relations between different entities. Specifically, this paper defines the strength parameter between different relations and same relations in a neural network when dealing with different entities,

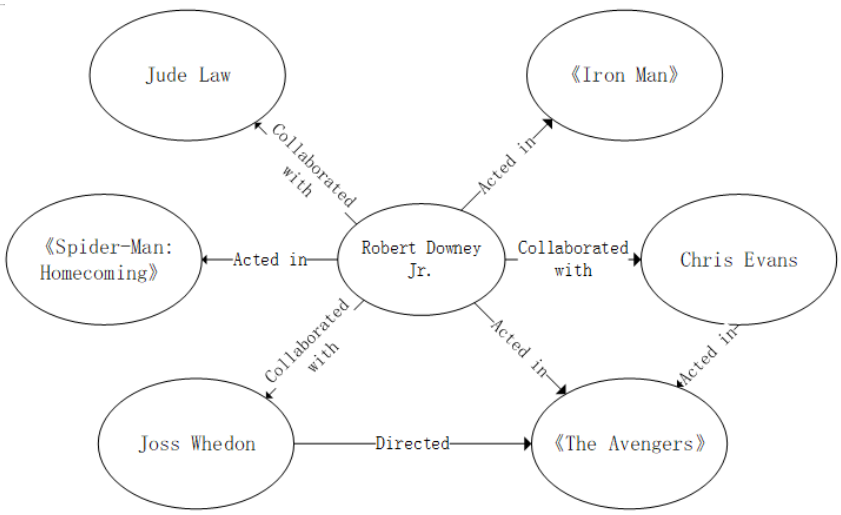

Fig. 1. Example of a partial knowledge graph for film actor Robert Downey Jr

in order to obtain from this parameter the information about the relations that are present in neighboring nodes and edges. The contributions of this paper are summarized below:

This paper proposes a framework with RS-GCN as the encoder and RotatE as the decoder. The encoder side of the framework is based on GCN, and in contrast to traditional GCN-based models, RS-GCN utilizes graph structure and aggregates the strength information of neighboring entities and relations between entities and incorporates it into knowledge graph complements;

In the link prediction task, RS-GCN has excellent performance on multiple datasets, and it also outperforms its baseline, verifying the validity and feasibility of RS-GCN. For example, on the FB15k-237 dataset, there is a 5.6% improvement in MRR over its baseline RotatE, and 6.2%, 2.4%, and 3.7% on Hits@1, Hits@3, and Hits@10, respectively.

2. Related Works

In the knowledge graph completion task. TransE [7] is a simple and effective learning model for knowledge graph representation, but it has some limitations in dealing with symmetry, many-to-many relations, and sparsity. In order to solve the problem, TransE-based extensions such as TransH [15],TransR [8] and STransE [16] have been proposed.These extensions take into account the type information of the relations in order to realize the mapping of the head-tail entities in the relational space through the introduction of hyperplanes, relation-specific mapping matrices, and bilinear transformations, and so on. In addition, RotatE [17] is able to better capture relations between entities, especially symmetric relations, by introducing complex vectors and rotation operations, In real application scenarios, the translation model cannot adapt to the knowledge graph application environment, where the data is huge, complex and constantly changing.Some researchers factor in the dynamics,.Semantic matching models such as DistMult [11], ComplEx [12], SimplE [18], and QuatE [19] are usually based on similarity metrics to evaluate the semantic similarity between entities or relations. In addition, neural network based models such as ConvE [9], ConvKB [10] and CapsE [20] have been used to learn the embedding of entities and relations within triples, however, the neural network models consider each triple as a whole and do not take into account the relations between the triples. However, these embedding models do not consider entity-to-entity message passing and semantic information. Especially when applied to those knowledge graphs where the entities themselves have fewer attribute descriptions but richer global information, the expressive power of these models is significantly limited.

Some researches aim at modeling the information of knowledge structure by considering the knowledge graph as a special kind of graph structure. The GNN-based model follows the principle of neural message passing, which achieves the learning of the global representation of the whole graph by iteratively aggregating the local information of the nodes, and then efficiently captures the structural and topological features of the graph. The TransGCN [21] model combines the translation model with GCN, synthesizes the mapping of entities and relations by TransE model and the extraction of features by graph neural network, and achieves a significant improvement in the results of the link prediction task. Neighborhood information is also effectively used for modeling in models such as RGCN [13] and SACN [14], where the encoder model accumulates evidence in multiple inference steps

in the graph, taking into account the effect of differences in relations in the knowledge graph on entity embedding. However, most of the above approaches construct knowledge models based on the structural information of the knowledge graph and the semantic information of the entities, or consider only several types of relations and lack of semantic information utilization of the same relations between different entities. On this basis, this paper's approach further considers and exploits the potential semantic information of neighbor relations to improve the expressive power and complementary accuracy of the model.

3. Method

In this section, we specifically describe our proposed RS-GCN. the general framework of the model is given in Fig. 2, which contains both encoder and decoder ends: (1) The encoder side, built on GCN, is used to better aggregate the information carried by nodes and edges. While the graph convolutional network aggregates the features of neighboring nodes, RS-GCN further enhances the representation of the central entity by constructing a weight matrix between different entities under each relation. An example of the weight matrix Θ is shown in Fig. 3. (2) On the decoder side, RotatE is integrated into the model as a decoder. The RotatE model is a more advanced model that can effectively model three relational patterns and predict missing entities and relations.

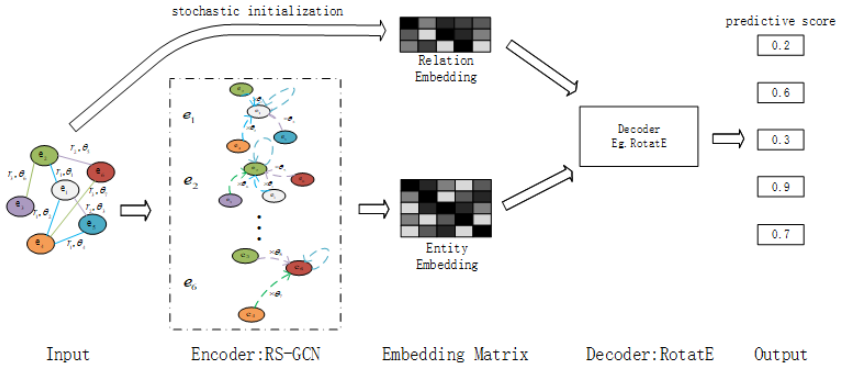

Fig. 2. Overall framework flowchart

3.1. Encoder

The encoder module is an extension of the classical GCN as it weighs different types of relations in different ways during aggregation and learns the weights adaptively during network training. Introducing the following notation: a directed and labeled multigraph is denoted as $G = (V, E, R)$, where nodes (entities) $v_i \in V$ and labeled edges (relations) $(v_i, r, v_j) \in E$, where $r \in R$ is a relation type. The related methods can be concretely represented as:

$$h_i^{(l+1)} = \sigma(\sum_{j \in N_i} g(h_i^{(l)}, h_j^{(l)})). \tag{1}$$

Where $h_i^{(l)} \in R^{d^{(l)}}$ is the hidden state of the node v_i in the $l-th$ layer of the neural network and $d^{(l)}$ is the dimension of the layer representation. The form $g(\cdot)$ is usually chosen as a function of a (message-specific) neural network that continuously accumulates incoming information, which is passed through an elemental activation function $\sigma(\cdot)$, such as the ReLU function. N_i denotes the set of direct neighbors of node v_i. Aggregation of neighboring nodes of node v_i is denoted. R-GCN is inspired to redefine the forward pass update of entities or nodes by considering the different relations represented by edges in the graph structure:

$$h_i^{(l+1)} = \sigma(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}). \tag{2}$$

Where N_i^r denotes the set of direct neighbors of node v_i under relation $r \in R$, and $c_{i,r}$ is a problem-specific normalization constant that can be learned or chosen in advance (e.g., $c_{i,r} = |N_i^r|$).

Referring to the formula above, R-GCN utilizes information from neighboring distinct relations to enhance the representation of the central entity. Indeed, even if the relations are the same, the same relation between different

entities is not equally important in indicating the central entity. For example, in Fig.1 presented earlier, when the same relation r is "Acted in", it is clear that special appearances, sole lead actor and multiple lead actors have different relevance and indications between different entities. Similarly, when the relation r is "Collaborated with", collaboration with the director and collaboration with the actors naturally have different relevance and indications.

Therefore, this paper constructs a weight matrix between different entities under each relation Θ further enhancement of semantic feature interaction during entity encoding. Relations between different entities possess different weights which are distinguished by the learnable parameter $\theta_m, 1 \leq m \leq M$ and M denotes the number of edges in the graph structure. Thus, the formula can be expressed as follows:

$$h_i^{(l+1)} = \sigma(\sum_{j \in N^i} \theta_m^{(l)} h_j^{(l)} W^{(l)} + W_0^{(l)} h_i^{(l)}). \tag{3}$$

θ_m represents the strength of the relationship in which the relation between this entity and another entity has, and also, considering the importance of normalization under the same relation, the weight matrix Θ as shown in Fig. 3,we specifies that in the matrix represented by the same relation, the sum of the values of θ_m in each row is 1 for normalization.

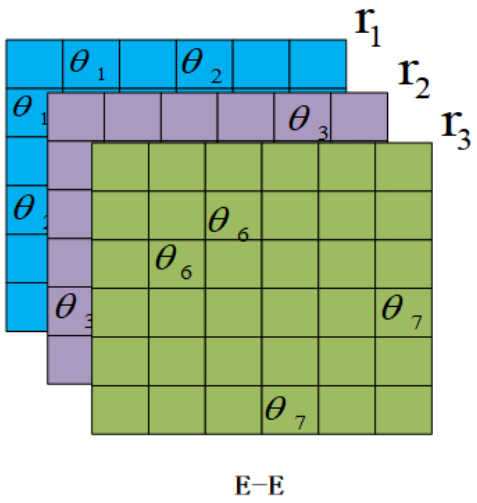

Fig. 3. Example of a weighting matrix (corresponding to the knowledge graph in Fig.2)

Overall, Eq. (3) conveys the feature information of neighboring nodes by assigning different relation strengths to the relations and by normalizing the sum. The choice of (Wh_j) node-to-node linear transformation a can give the model a computational advantage. Meanwhile, noting that the weight matrix Θ with intensity information in the matrix corresponding to each relation θ_m is irregularly dispersed and has a large number of elements that are zero, In order to improve computational efficiency, a sparse matrix representation of the data is used as a solution to efficiently store non-zero elements, capturing the inherent sparsity in the dataset. In addition, the model sets up a single self-connection at each node in the data to convey its own features so that the features of the nodes in the previous layer can be efficiently captured by the nodes in the next layer.

3.2. Decoder

Embedding-based models are often used as the basic model of a decoder, which enables the model to learn a corresponding embedding vector that maps entities into a low-dimensional space. This allows for closer representation of similar entities in the embedding space, better capturing of semantic associations in the knowledge graph, and effective prediction in the face of unseen entities and relations. The decoder takes as input the embedding of entities and relations from the encoder. We use RotatE as the decoder, which uses complex vectors to represent entities and relations, and introduces a rotation operation to capture the relations between entities, and is able to model three of the more common relational patterns, formulated as follows.

For any entities a, b, the relation r is symmetric (antisymmetric):

$$r(a,b) \rightarrow r(b,a), r(a,b) \rightarrow \neg r(b,a). \tag{4}$$

For any entities a, b, the relation r_1 is inverse to the relation r_2.

$$r_2(a,b) \rightarrow r_1(b,a). \tag{5}$$

For any entity a, b, c, a relation r_1 is a combination of relation r_2 and relation r_3.

$$r_2(a,b) \wedge r_3(b,c) \rightarrow r_1(a,c). \tag{6}$$

Given (v_i, r_k, v_j) ternary, the ternary scores in RotatE:

$$\psi_{r_k}(v_i, v_j) = -||v_i \odot r_k - v_j||. \tag{7}$$

$\odot$ is the product of elements in complex space.

To be consistent with the baseline RotatE, the model uses self-adversarial negative sampling to train the model. The main feature of self-adversarial negative sampling is that negative triples are assigned different probabilities, apparently wrong negative triples are assigned lower probabilities and high quality negative triples are assigned higher probabilities. Thus, the probability distribution p is used to draw negative samples according to the current embedding model, for the triad (h, r, t).

$$p(h'_j, r, t'_j|(h_i, r_i, t_i)) = \frac{e^{\alpha\psi_r}(h'_j, t'_j)}{\sum_i e^{\alpha\psi_r}(h'_i, t'_i)}. \tag{8}$$

Where α is the sampled temperature,the scores of negative examples are amplified by the exponential function exp, which enhances the difference between positive and negative examples, making it easier for the model to distinguish between positive and negative examples during training. The above probabilities for negative samples are then considered as the weights of the samples to help construct the loss function. The loss function can be written as follows:

$$L = -\log(\sigma(\gamma + \psi_r(h,t))) - \sum_{i=1}^{n} p(h'_i, r, t'_i) \log(\sigma(-\psi_r(h'_i, t'_i) - \gamma)). \tag{9}$$

σ is a sigmoid function, and all embeddings are in the complex vector space.

4. Experiments and Analysis

4.1. Dataset

In previous studies, the performance of link prediction methods is usually evaluated on four datasets, namely FB15k [3], FB15k-237 [22], WN18 [23] and WN18RR [22]. Tab.1 shows the basic statistics of these four datasets. FB15k is a dataset extracted from the original FreeBase Knowledge Graph for the knowledge graph representation learning task, and WN18 is a dataset extracted from the original WordNet English vocabulary database for the knowledge graph representation learning task.According to the literature [9], the presence of inverse triples in both the training and test data leads to an effect on the prediction by memorizing these affected triples on the predictive ability to have an impact. To address this issue, FB15k-237 and WN18RR are proposed as variants of the above two datasets, where all inverse ternary pairs are removed. These two datasets have been shown to be more challenging for models performing link prediction.

Table 1. Statistical information on data sets

Dataset	$\|E\|$	$\|R\|$	#Triples(Train)	#Triples(Valid)	#Triples(Test)
FB15k	14951	1345	483142	50000	59071
FB15k-237	14541	237	272115	17535	20466
WN18	40943	18	141442	5000	5000
WN18RR	40943	11	86835	3034	3134

4.2. Evaluation Protocol

In the testing phase, for each ternary, this paper replaces the head entity with all other entities in the current KG and computes the scores of these replaced ternaries and the original ternary using the scoring function specified in Section 3. Following previous studies, using the most commonly used metrics Mean Reverse Rank (MRR) and Hits@n $(n = 1, 3, 10)$. For all metrics, higher values imply better performance [26-29].

4.3. Training Settings

Models uses PyTorch [24] as a deep learning framework, and all experimental operations were performed on an A40 graphics card, using Adam as a gradient optimizer. The best parameters are selected when the filtered MRR achieves the best performance on each validation set. Some of the parameters are chosen as follows:the number of samples per training batch_size is 1024, the learning rate lr is 0.0001, and the number of samples per test test_batch_size is 16.

4.4. Experimental Results and Analysis

The experimental results of the RS-GCN model and other comparative models on the link prediction task are shown in Tab.2 and Tab.3, in which some of the experimental results of the advanced models are referred to other literatures [25], and the individual experimental results are reproduced due to the unknown source. The model sets the max_step of RS-GCN to 100000 under the same environment settings and repeats the experiment three times under the optimal parameters, and takes the average value of each evaluation index as the experimental results. Tables highlights the best performance after comparison in bold.

On the FB15k-237 and WN18RR datasets, RS-GCN outperforms existing classical models on all metrics, showing competitive results. These two datasets mainly contain various inference modes such as symmetric/antisymmetric and combinatorial. Among these modes, TransE has some limitations in dealing with symmetry, many-to-many relations, and sparsity, while RotatE improves inference by transforming relations into rotations in complex space, which gives better performance of the RotatE model compared to other methods in Euclidean KGEs. In contrast, RS-GCN adopts the encoder-decoder framework, considers the internal strength of the same relation and different relations among different entities, and combines the advantages of the RotatE model, which exhibits high performance after satisfying the normalization constraints, demonstrating its high representational capability.

In contrast, the performance of RS-GCN on the WN18 dataset is not very different from that of the advanced classical models (e.g., ConvE, NagE, and TuckER, etc.), and it can be found that the WN18 dataset does not provide a significant performance differentiation between these models within the error tolerance. In the FB15k and WN18 datasets, since the datasets mainly contain multiple inference modes such as symmetric/antisymmetric and inverse, the performance of the TransE model is relatively poor because TransE cannot handle symmetric modes. However, RS-GCN is able to handle these patterns better and achieves similar performance without excessive parameters compared to models such as TuckER.

Overall, RS-GCN shows competitive performance in handling symmetric/antisymmetric and combinatorial inference patterns and achieves satisfactory results on different datasets.

5. Conclusion

This paper proposes an encoder-decoder framework (RS-GCN) to simultaneously utilize the information provided by edges in a graph structure. When dealing with identical relations between different entities, proposing a new approach to distinguish different strengths of identical relations by introducing relationship weights (stored as matrices). The model likewise considers normalizing constraints within the same relation to distinguish between different relations. The experimental results for the four datasets FB15k, WN18, FB15k-237 and WN18RR show that our RS-GCN model consistently outperforms the baseline R-GCN and RotatE models in all metrics, which proves the validity of considering the strength of the relationship. In addition, although neighbor information is encoded and learned in the RS-GCN framework, there is still a lot of auxiliary information such as textual information and attribute information that is overlooked. In the future, we plans to explore ways to utilize other auxiliary information so that the easily ignored auxiliary information of knowledge graphs can also play a role in relational embedding learning.

6. Conflict of Interest

The authors declare that there are no conflict of interests, we do not have any possible conflicts of interest.

Acknowledgments. None.

Table 2. Link prediction results on FB15k-237 and WN18RR

Models	FB15k-237(MRR)	FB15k-237(Hits@1)	FB15k-237(Hits@3)	FB15k-237(Hits@10)	WN18RR(MRR)	WN18RR(Hits@1)	WN18RR(Hits@3)	WN18RR(Hits@10)
TransE	0.294	none	none	0.465	0.226	none	none	0.501
DistMult	0.241	0.155	0.263	0.419	0.43	0.39	0.44	0.49
ComplEx	0.247	0.158	0.275	0.428	0.44	0.41	0.46	0.51
R-GCN	0.248	0.153	0.258	0.417	0.441	0.360	0.485	0.492
ConvE	0.325	0.237	0.356	0.501	0.43	0.40	0.44	0.52
RotatE	0.338	0.241	0.375	0.533	0.476	0.428	0.492	0.571
NagE	0.340	0.244	0.378	0.530	0.477	0.432	0.493	0.574
RS-GCN	0.357	0.256	0.384	0.553	0.488	0.439	0.507	0.577

Table 3. Link prediction results on FB15k and WN18

Models	FB15k(MRR)	FB15k(Hits@1)	FB15k(Hits@3)	FB15k(Hits@10)	WN18(MRR)	WN18(Hits@1)	WN18(Hits@3)	WN18(Hits@10)
TransE	0.463	0.297	0.578	0.749	0.495	0.113	0.888	0.943
DistMult	0.654	0.546	0.733	0.824	0.822	0.728	0.914	0.936
ComplEx	0.692	0.599	0.759	0.840	0.941	0.936	0.936	0.947
R-GCN	0.696	0.601	0.760	0.842	0.814	0.697	0.929	0.964
ConvE	0.657	0.558	0.723	0.831	0.943	0.935	0.946	0.956
RotatE	0.797	0.746	0.830	0.884	0.949	0.944	0.952	0.959
TuckER	0.795	0.741	0.833	0.892	0.953	0.949	0.955	0.958
RS-GCN	0.815	0.752	0.844	0.897	0.951	0.945	0.956	0.958

References

1. Ji S, Pan S, Cambria E, et al. A survey on knowledge graphs: Representation, acquisition, and applications[J]. IEEE transactions on neural networks and learning systems, 2021, 33(2): 494-514.
2. Auer S, Bizer C, Kobilarov G, et al. Dbpedia: A nucleus for a web of open data[C]//international semantic web conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 722-735.
3. Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008: 1247-1250.
4. Hamilton W, Bajaj P, Zitnik M, et al. Embedding logical queries on knowledge graphs[J]. Advances in neural information processing systems, 2018, 31.
5. Mai G, Yan B, Janowicz K, et al. Relaxing unanswerable geographic questions using a spatially explicit knowledge graph embedding model[C]//Geospatial Technologies for Local and Regional Development: Proceedings of the 22nd AGILE Conference on Geographic Information Science 22. Springer International Publishing, 2020: 21-39.
6. Ying R, He R, Chen K, et al. Graph convolutional neural networks for web-scale recommender systems[C]//Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 974-983.
7. Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. Advances in neural information processing systems, 2013, 26.
8. Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]//Proceedings of the AAAI conference on artificial intelligence. 2015, 29(1).
9. Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2d knowledge graph embeddings[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).
10. Nguyen D Q, Nguyen T D, Nguyen D Q, et al. A novel embedding model for knowledge base completion based on convolutional neural network[J]. arxiv preprint arxiv:1712.02121, 2017.
11. Yang B, Yih W, He X, et al. Embedding entities and relations for learning and inference in knowledge bases[J]. arxiv preprint arxiv:1412.6575, 2014.
12. Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction[C]//International conference on machine learning. PMLR, 2016: 2071-2080.
13. Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks[C]//The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3C7, 2018, proceedings 15. Springer International Publishing, 2018: 593-607.
14. Shang C, Tang Y, Huang J, et al. End-to-end structure-aware convolutional networks for knowledge base completion[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 3060-3067.
15. Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]//Proceedings of the AAAI conference on artificial intelligence. 2014, 28(1).
16. Nguyen D Q, Sirts K, Qu L, et al. Stranse: a novel embedding model of entities and relationships in knowledge bases[J]. arxiv preprint arxiv:1606.08140, 2016.
17. Sun Z, Deng Z H, Nie J Y, et al. Rotate: Knowledge graph embedding by relational rotation in complex space[J]. arxiv preprint arxiv:1902.10197, 2019.
18. Kazemi S M, Poole D. Simple embedding for link prediction in knowledge graphs[J]. Advances in neural information processing systems, 2018, 31.
19. Zhang S, Tay Y, Yao L, et al. Quaternion knowledge graph embeddings[J]. Advances in neural information processing systems, 2019, 32.
20. Nguyen D Q, Vu T, Nguyen T D, et al. A capsule network-based embedding model for knowledge graph completion and search personalization[J]. arxiv preprint arxiv:1808.04122, 2018.
21. Cai L, Yan B, Mai G, et al. TransGCN: Coupling transformation assumptions with graph convolutional networks for link prediction[C]//Proceedings of the 10th international conference on knowledge capture. 2019: 131-138.
22. Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference[C]//Proceedings of the 3rd workshop on continuous vector space models and their compositionality. 2015: 57-66.
23. Miller G A. WordNet: a lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39-41.
24. Van Merri nboer B, Breuleux O, Bergeron A, et al. Automatic differentiation in ML: Where we are and where we should be going[J]. Advances in neural information processing systems, 2018, 31.
25. Lin X, E Haihong, Song W, Luo H, Song M. QubitE:Qubit Embedding for Knowledge Graph Completion[J]. Computer Science, 2023, 50(11): 201-209.
26. Yin S, Li H, Sun Y, et al. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactions on Science and Engineering, 2024, 2(2): 13-20.
27. Teng L, Qiao Y, Shafiq M, et al. FLPK-BiSeNet: Federated learning based on priori knowledge and bilateral segmentation network for image edge extraction[J]. IEEE Transactions on Network and Service Management, 2023, 20(2): 1529-1542.
28. Zhao Y, Li H, Yin S. A multi-channel character relationship classification model based on attention mechanism[J]. International Journal of Mathematical Sciences and Computing, 2022, 8: 28-36.
29. Teng L, Li H, Yin S, et al. An active contour model based on hybrid energy and fisher criterion for image segmentation[J]. International Journal of Image and Data Fusion, 2020, 11(1): 97-112.

Biography

Mingshao Xu is with the Software College, Shenyang Normal University. Research direction is computer application and AI.

Hang Li is with the Software College, Shenyang Normal University. Research direction is computer application and AI.

Zhi Hu is with the Software College, Shenyang Normal University. Research direction is computer application and AI.