Link Prediction based on Tensor Decomposition for the Knowledge Graph of COVID-19 Antiviral Drug Postprint

Authors: Ting, Jia, Yuxia Yang, Xi, Lu, Qiang, Zhu, Yang Kuo, Xuezhong, Zhou, Kuo, Yang, Xuezhong Zhou

Date: 2022-11-28T00:00:00+00:00

Abstract

Due to the large-scale spread of COVID-19, which has a significant impact on human health and social economy, developing effective antiviral drugs for COVID-19 is vital to saving human lives. Various biomedical associations, e.g., drug-virus and viral protein-host protein interactions, can be used for building biomedical knowledge graphs. Based on these sources, large-scale knowledge reasoning algorithms can be used to predict new links between antiviral drugs and viruses. To utilize the various heterogeneous biomedical associations, we proposed a fusion strategy to integrate the results of two tensor decomposition-based models (i.e., CP-N3 and ComplEx-N3). Sufficient experiments indicated that our method obtained high performance (MRR=0.2328). Compared with CP-N3, the mean reciprocal rank (MRR) is increased by 3.3% and compared with ComplEx-N3, the MRR is increased by 3.5%. Meanwhile, we explored the relationship between the performance and relationship types, which indicated that there is a negative correlation (PCC=0.446, P-value=2.26e-194) between the performance of triples predicted by our method and edge betweenness.

Full Text

Preamble

Link Prediction based on Tensor Decomposition for the Knowledge Graph of COVID-19 Antiviral Drug

Ting Jia[1], *Yuxia Yang*[1], Xi Lu[1], Qiang Zhu[1], Kuo Yang[1],[2]† & Xuezhong Zhou[1],†

[1]Institute of Medical Intelligence, School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

[2]BNRIST/Department of Automation, Tsinghua University, Beijing 100084, China

Keywords: Link prediction; Knowledge graph; COVID-19; Antiviral drug prediction; Tensor decomposition

Citation: Jia, T., et al.: Link prediction based on tensor decomposition for the knowledge graph of COVID-19 antiviral drug. *Data Intelligence* 4(1), 134-148 (2022). doi: 10.1162/dint_a_{00117}

Received: February 14, 2021; **Revised:** June 10, 2021; **Accepted:** August 17, 2021

Abstract

The large-scale spread of COVID-19 has significantly impacted human health and the global economy, making the development of effective antiviral drugs crucial for saving lives. Various biomedical associations—such as drug-virus and viral protein-host protein interactions—can be used to construct biomedical knowledge graphs. Based on these resources, large-scale knowledge reasoning algorithms can predict novel links between antiviral drugs and viruses. To leverage diverse heterogeneous biomedical associations, we propose a fusion strategy that integrates results from two tensor decomposition-based models (CP-N3 and ComplEx-N3). Extensive experiments demonstrate that our method achieves high performance (MRR=0.2328), improving MRR by 3.3% over CP-N3 and 3.5% over ComplEx-N3. Furthermore, we investigated the relationship between prediction performance and relationship types, revealing a negative correlation (PCC=0.446, P-value=2.26e-194) between the performance of triples predicted by our method and edge betweenness.

1. Introduction

A knowledge graph (KG) is a structured representation of real-world information, where nodes represent entities (e.g., people, places), edges represent specific facts connecting two entities, and labels denote edge types [1, 2]. As a foundation for knowledge engineering applications, KGs play a crucial role in various artificial intelligence tasks (e.g., clinical decision-making and question-answering systems). However, even state-of-the-art KGs suffer from incompleteness [1, 3, 4], including Freebase [5], Wikidata [6], DBpedia [7], and Google KG [8]. Link prediction (LP), which leverages existing facts in a KG to infer missing ones, represents one of the most promising approaches to address this problem [8].

Amid the large-scale spread of COVID-19 [9], the China Conference on Knowledge Graph and Semantic Computing (CCKS) 2020 organized a link prediction task for the COVID-19 Antiviral Drug Knowledge Graph (ADKG), requiring participants to predict associations between biomedical entities. This task holds

significant importance and academic value for antiviral drug research and development.

Current approaches to link prediction include node attribute similarity, network structure analysis, likelihood-based methods, and machine learning techniques. Among these, some employ classification methods such as decision trees [10], support vector machines [11], and multi-layer perceptrons [12], while others utilize probabilistic graphical models and matrix factorization methods [13, 14, 15, 16]. In recent years, tensor decomposition has attracted considerable attention [14]. Canonical Tensor Decomposition (CP) decomposes a high-order tensor into several one-dimensional factor matrices [13, 14, 15, 16, 17]; for example, a 3rd-order tensor can be decomposed into three one-dimensional factors, representing a low-rank approximation or feature extraction of high-dimensional data. ComplEx [16], a complex-valued representation method, can capture both symmetric and asymmetric relationships in binary KG relations. However, designing effective strategies for fusing different link prediction algorithms remains challenging. Furthermore, it remains unclear how different relationship types and even individual test triples exhibit varying performance in link prediction algorithms.

Therefore, we introduced ensemble learning principles to the link prediction task. We adopted ensemble methods to combine multiple supervised models, yielding a more robust and comprehensive predictive model. Participating in the CCKS 2020 link prediction task, we proposed a fusion strategy that effectively integrates results from two tensor decomposition models (CP-N3 and ComplEx-N3). Experimental results demonstrated that our method achieved high performance (MRR=0.2328), ranking third in the competitive CCKS 2020 Challenge Task 1. Concurrently, to investigate whether prediction performance correlates with relationship types in real-world data, we analyzed results across different relationship types in the benchmark dataset. From a complex network perspective, we examined the relationship between prediction results for different entities and edge betweenness, an important edge centrality metric.

2. Related Work

Generally, two approaches exist for link prediction on KGs [18]: embedding-based methods [16, 19, 20] and rule-based methods [18, 21, 22]. Embedding-based methods map entities and relations in KGs to low-dimensional vector spaces. Rule-based methods attempt to capture inherent regularities in KGs as rules [18] and then apply these learned rules for reasoning.

TransE [19] is a classical translational distance model that treats relations as translations from head to tail entities. Inspired by TransE, many variants have been proposed, such as STransE [23] and RotatE [20]. With advances in deep learning, neural network-based models have been increasingly applied to LP tasks, employing neural networks to compute scoring functions for triples. ConvE [24] is a multi-layer convolutional network designed to extract higher-

level, non-linear features. R-GCN [25] was developed specifically to handle the highly multi-relational nature of KGs. RSN [26] combines RNNs with residual learning to learn relation paths, effectively capturing long-term relational dependencies.

In recent studies, tensor decomposition-based methods have attracted considerable attention [13, 14]. Tensor decomposition-based representation learning treats each triple as an element in a tensor and performs representation learning via tensor decomposition algorithms. In these methods, the tensor is composed of low-dimensional vectors that serve as embeddings for entities and relationships. RESCAL [27], the first typical algorithm using tensor decomposition for KG completion, employs a 3rd-order tensor to represent entities and relationships. However, due to its large number of parameters, RESCAL is prone to overfitting, leading to the development of DistMult [15]. While reducing the parameter space, DistMult [15] introduces a new limitation: it treats all relations as symmetric. ComplEx [16] extends DistMult into the complex domain, using the Hermitian dot product to compute triple scores and probability to determine triple validity, enabling it to model asymmetric relationships. CP [13] frames link prediction as a 3rd-order binary tensor completion problem, embedding head and tail entities independently.

Beyond embedding-based methods, rule-based approaches have also received significant attention. Some path-based methods use sets of paths generated through random walks as input. MINERVA [22] eliminates the need for precomputed paths by training a reinforcement learning agent to walk to answer nodes conditioned on input queries. RNNLogic [21] learns logic rules for KG reasoning, treating logic rules as latent variables and simultaneously training a rule generator and a reasoning predictor. EARDict [18] provides a foundational theory for KG reasoning based on ending-anchored rules, generating numerous rules to evaluate the validity of unknown triples.

3.1 Data Preprocessing

First, we used regular expressions to parse the official data files and extract triples in the required format. Second, we incorporated entity attribute lists into our knowledge graph to enrich entity information. Finally, to expand relationships in the knowledge graph, we applied a well-known data augmentation technique [13, 24] that adds reciprocal relations for every triple, i.e., adding (t, r^{-1}, h) for every (h, r, t), where h is the head entity, r is the relation, and t is the tail entity.

3.2 CP-N3

CP [17] represents a tensor as a sum of R rank-one tensors u_r^(1) u_r^(2) u_r^(3) (denotes tensor product) where r {1, ···, R}. The KG completion performance of standard CP on benchmarks lags behind its competitors. Timothee et al. [13] proposed a novel regularizer based on tensor nuclear p-norms

and presented a reformulated problem that enables CP decomposition to achieve state-of-the-art performance on several datasets (named CP-N3).

3.3 ComplEx-N3

Similar to DistMult [15], ComplEx [16] represents each relation type as a diagonal matrix but extends the vector space into the complex domain: h ^d, t ^d, and r ^(d×d). In the complex domain, the bilinear product becomes a Hermitian product, where the conjugate-transpose t* is used instead of the traditional transpose. This allows ComplEx to successfully model asymmetric relations. Timothee et al. [13] proposed a ComplEx-based model (named ComplEx-N3) with the novel regularizer and systematic parameter searches, achieving better performance than ComplEx.

3.4 A Fusion Strategy

Unlike traditional machine learning methods that train a single learner, ensemble learning methods train multiple learners and combine their predictions to achieve superior results. These methods have been successfully applied to various fields, including computer vision [28], species distribution prediction [29], and auxiliary diagnosis [30].

Inspired by ensemble learning principles, we designed a fusion strategy for this task (Figure 1 [Figure 1: see original paper]). First, we use the same KG to train the CP-N3 and ComplEx-N3 models separately. Second, we use each model to generate predictions and select the top-20 entities with the highest scores. Third, we assign weights from 20 to 1 based on entity rankings (e.g., rank 1 receives weight 20, rank 2 receives weight 19, etc.). Finally, we integrate the two top-20 lists through weighted voting to obtain the final top-10 predictions. During integration, we filter entities based on type, eliminating those that do not match the expected entity type for the given relation.

4.1 Datasets

We used the ADKG from CCKS 2020 to evaluate our approach. Published for CCKS 2020 Task 1, this dataset includes three entity types (proteins, viruses, and drugs) and four relation types (effect, interaction, produce, and binding). ADKG contains 7,844 entities and 40,000 triples. Additionally, two prediction batches were provided: the first containing 4,256 triples and the second containing 5,000 triples. We refer to these as prediction sets because each triple lacks either its head or tail entity, i.e., (h, r, ?) or (?, r, t). Our task is to predict the missing entities based on the trained model.

To evaluate model performance, we randomly selected 10% of each relation type as the test set, resulting in 36,000 training triples and 4,000 test triples. To enrich entity attribute information, we added entity attribute lists to the training set and simultaneously incorporated the involved entities into the attribute en-

tity list. In total, the knowledge graph contains 8,494 entities and 51,131 triples, with 47,131 triples in the training set and 4,000 in the test set.

4.2 Experimental Setup

We tuned all hyperparameters based on predictive performance on the test set to identify optimal settings for ADKG. Table 1 shows the best parameter settings for both methods. In our experiments, we used these optimal parameters to generate top-20 predictions from each method and integrated them using our fusion strategy to produce final results.

Table 1. Best parameter settings of CP-N3 and ComplEx-N3 on ADKG.

Hyper-parameters	CP-N3	ComplEx-N3
Dimension		
Learning rate		
Batch size		
Regularization coefficient		
Initialization		
Optimizer	Adagrad	Adagrad

5.1 Comparison Results with Baselines

For each test triple, we generated 8,494 candidate triples by combining the test entity-relation pair with all possible entities E, ranking the resulting scores. We employed the filtered setting [19], where all known true triples are removed from the candidate set except for the entity being evaluated in the current test triple. Although only the top-10 candidates were submitted for evaluation, we considered predictions beyond the top 10 for effective result integration. Therefore, we used mean reciprocal rank of top-10 (MRR@10) and Hits@k (k {10, 20, 50, 100}) as evaluation metrics. MRR@10 is the average of inverse ranks over the top-10 candidate triples, while Hits@k measures the percentage of true triples ranked within the top-k candidates.

Link prediction performance comparisons on the ADKG test set are presented in Table 2 . In addition to the two provided baselines (TransE [19] and R-GCN [25]), we evaluated six other classic methods: DistMult [15], ComplEx [16], TuckER [14], HypER [31], CrossE [32], and ConvE [24]. Results show that CP-N3 (MRR@10=0.185) and ComplEx-N3 (MRR@10=0.183) outperformed not only linear models (e.g., DistMult, ComplEx, TuckER) but also deep neural network models (e.g., R-GCN, HypER, CrossE, ConvE). Consequently, we selected these two methods as our base learners and integrated their predictions using our fusion strategy (MRR@10=0.218). In the competitive CCKS 2020 Challenge Task 1, our method achieved a score of 19.785 (ranking first) on the first prediction batch and 23.275 (ranking third) on the second batch, where MRR@10$×$100 serves as the evaluation metric.

Table 2. Comparison results of link prediction methods.

Model	Hits@10	Hits@20	Hits@50	Hits@100	MRR@10
TransE [19]					
R-GCN [25]					
DisMult [15]					
ComplEx [16]					
TuckER [14]					
HypER [31]					
CrossE [32]					
ConvE [24]					
ComplEx-N3 [13]					0.183
CP-N3 [13]					0.185
Fusion Strategy (ours)					0.218

5.2 Parameter Sensitivity

To investigate the impact of hyperparameters on CP-N3 and ComplEx-N3, we systematically tuned all relevant hyperparameters to identify optimal settings for ADKG. We used Hits@k (k {10, 20, 50, 100}), MRR, and MRR@10 as evaluation metrics. Four key parameters were tuned: (1) embedding dimension for entities and relations, tuned over {500, 1000, 1500, 2000, 2500}; (2) learning rate, controlling parameter update magnitude during optimization, tuned over {0.05, 0.06, 0.07, 0.08, 0.09, 1.0}; (3) batch size, determining the number of samples trained simultaneously, tuned over {128, 256, 512, 1024}; and (4) regularization coefficient, tuned over {0.005, 0.01, 0.05, 0.1}. We conducted grid search by fixing three parameters at a time while tuning the fourth.

Experimental results are shown in Figure 2 [Figure 2: see original paper] (A-D for CP-N3 and E-H for ComplEx-N3). For CP-N3, both small (128) and large (512 or 1024) batch sizes yielded poor performance, while a batch size of 256 achieved optimal results. Embedding dimensions between 1000 and 2500 had minimal impact, and small adjustments to the learning rate showed little effect, demonstrating the model' s strong robustness in candidate entity prediction. For ComplEx-N3, optimal performance was achieved with a batch size of 128, dimension of 2500, regularization coefficient in {0.005, 0.01, 0.1}, and learning rate in {0.07, 0.09}.

To investigate the impact of fusion size on performance, we conducted experiments with different Top-n values (ranging from {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}). Results in Figure 3 [Figure 3: see original paper] show that prediction performance initially improves with increasing fusion size, with Hits@10 peaking at Top-30 and MRR at Top-20. However, performance declines beyond Top-50 to varying degrees. Considering both metrics, we selected Top-20 as our final fusion size.

Figure 2. The prediction performance with tuning the hyper-parameters. The first parameter is the entity and relation dimension of the embedding vectors, tuning from {500, 1000, 1500, 2000, 2500}. The second parameter is learning rate, tuning from {0.05, 0.06, 0.07, 0.08, 0.09, 1.0}. The third parameter is batch size, tuning from {128, 256, 512, 1024}. The final parameter is regularization coefficient, tuning from {0.005, 0.01, 0.05, 0.1}. A-D sub-figures are for CP-N3 and E-H sub-figures are for ComplEx-N3.

Figure 3. The prediction performance with fusing different top n values. The left sub-figure is the result of hits@10. It has a turning point at Top 20, reaches the optimal value at Top 30, and then stabilizes. The right sub-figure is the result of MRR. There is a turning point at Top 20 and it reaches the optimal value; it remains stable until turning and falling at Top 50, and then it stabilizes.

5.3 Result Analysis of Different Relationship Types

To comprehensively evaluate our algorithm, we analyzed its performance across different relationship types (Table 3 & Figure 4 [Figure 4: see original paper]). Statistical results indicate that the binding relationship type achieves the best performance, followed by interaction, while effect shows the worst performance (Table 3). The binding type performs best for head entity prediction, interaction shows relatively balanced performance for both head and tail entity prediction, and effect exhibits the lowest overall performance (Figure 4).

The number of training triples likely influences prediction performance. For example, the effect type, with the least data, obtains the lowest performance. However, despite having fewer triples than the interaction type, binding achieves better head entity prediction. We analyzed the head and tail entity types and found that binding relationships always have virus proteins as head entities and host proteins as tail entities. In contrast, interaction relationships involve two distinct scenarios: virus protein → host protein and host protein → virus protein. This bidirectionality may increase prediction difficulty for interaction relationships.

Table 3. Comparison results of link prediction methods.

Relation type	Train	Head entity type	Tail entity type
Effect		Protein	Virus
Interaction		Protein/Virus	Virus/Protein
Produce		Protein	Virus
Binding		Virus	Protein

Additionally, to investigate performance variations across different triples, we constructed a heterogeneous network from all triples and computed edge betweenness centrality—a key edge metric in complex network analysis—for each

triple. We compared test triple rankings in prediction results with their betweenness values (Figure 5 [Figure 5: see original paper]). We analyzed head entity rankings, tail entity rankings, and their average, computing the Pearson correlation coefficient (PCC) for each. Results revealed a positive correlation between triple rankings and edge betweenness (PCCs = {0.406, 0.353, 0.446}, P-values = {2.272e-158, 1.158e-117, 2.256e-194}), indicating that triples located at the network center are more difficult to rank highly. Conversely, triples at the network periphery are easier to rank at the top.

Figure 4. Comparison of prediction performance of different relationship types. For the prediction of head entities, the binding relationship type achieved the best performance, although the number of the edges of this type is not the largest. For the prediction of tail entities, the interaction relationship type with the largest amount of data obtains the best performance. But overall, the results of interaction are relatively stable. The effect relationship type is the worst because of the very small amount of data.

Figure 5. Correlation analysis between prediction triples and edge betweenness. From left to right are the ranking results of the prediction head entity, the ranking result of the prediction tail entity, and the average result of the two rankings of 4,000 data records in the prediction set. The abscissa is the ranking, the ordinate is edge betweenness, the red line in the figure is the fitting curve of the blue scatter distribution, and each figure is marked with Pearson result and P-value.

5.4 Case Study

To illustrate our model' s predictive performance, we examined two triples from the first prediction batch—(?, interaction, ENTITY_{6303}) and (ENTITY_{3946}, binding, ?)—and generated their top-10 candidate entities (Table 4). Entities in bold represent true positives according to the benchmark. Rank1 shows ComplEx-N3' s top-10 predictions, Rank2 shows CP-N3' s top-10 predictions, and Rank3 shows the top-10 predictions from our fusion strategy.

For (?, interaction, ENTITY_{6303}), five of the top-10 candidates (ENTITY_{7358}, ENTITY_{5261}, ENTITY_{6303}, ENTITY_{2000}, and ENTITY_{1379}) were confirmed in the benchmarks. For (ENTITY_{3946}, binding, ?), four entities (ENTITY_{761}, ENTITY_{2386}, ENTITY_{6309}, and ENTITY_{3333}) were confirmed in benchmarks. This demonstrates that our model produces reliable integrated predictions.

Table 4. Top 10 candidate entities for two predictive triples.

(?, interaction, ENTITY_{6303})	(ENTITY_{3946}, binding, ?)
Candidate entities Rank1 Rank2 Rank3	**Candidate entities** Rank1 Rank2 Rank3
ENTITY_{7358}	ENTITY_{761}

(?, interaction, ENTITY_{6303})	(ENTITY_{3946}, binding, ?)
ENTITY_{5261}	ENTITY_{2386}
ENTITY_{6303}	ENTITY_{6309}
ENTITY_{3101}	ENTITY_{3333}
ENTITY_{2000}	ENTITY_{425}
ENTITY_{1379}	ENTITY_{224}
ENTITY_{4654}	
ENTITY_{2459}	
ENTITY_{5750}	
ENTITY_{7365}	

6. Discussion

In this study, we proposed an effective fusion strategy to integrate two tensor decomposition-based models, and experimental results demonstrated high performance on the KG link prediction task. We enriched our knowledge graph with entity attribute lists and expanded it by adding reciprocal relations [13, 24] for every triple. Nevertheless, the LP task still faces challenges, including designing KG embedding algorithms that incorporate richer ontology features and analyzing the equivalence between KG embedding and ontology reasoning.

Future research directions include: (i) combining logical rules containing rich background information with KG triples in a unified completion framework [33, 34]; and (ii) considering path relationships between triples in the knowledge graph [35]. In subsequent work, we will continue to explore new models, identify limitations of existing approaches, and improve performance.

Furthermore, our analysis of the relationship between prediction results and network attributes reveals their close connection, suggesting promising avenues for incorporating network attributes in future research.

Acknowledgements

This work was partially supported by Beijing Natural Science Foundation (No. M21012), National Key Research and Development Program of China (No. 2017YFC1703506, No. 2018AAA0100302) and National Natural Science Foundation of China (No. 82174533).

Author Contributions

T. Jia (tingjia@bjtu.edu.cn) and Y.X. Yang (yuxia__{yang}@163.com) conducted the experiments and contributed to manuscript writing. X. Lu (lxrlj123@163.com) administered the data collection and supervised the data input process. Q. Zhu (qzhu@bjtu.edu.cn) contributed to manuscript writing. K. Yang (yangkuo@bjtu.edu.cn) coordinated the writing of the whole manuscript. X.Z. Zhou (xzzhou@bjtu.edu.cn) contributed to the final version of

the manuscript. All authors have made meaningful and valuable contributions to revising and proofreading the manuscript.

References

[1] Rossi, A., et al.: Knowledge graph embedding for link prediction: A comparative analysis. arXiv preprint arXiv:2002.00819 (2020)
[2] Wang, Q., et al.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge & Data Engineering 29(12), 2724–2743 (2017)
[3] Lin, Y., et al.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2181–2187 (2015)
[4] Nathani, D., et al.: Learning attention-based embeddings for relation prediction in knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4710–4723 (2019)
[5] Bollacker, K., et al.: Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD ' 08), pp. 1247–1250 (2008)
[6] Vrandecic, D., Krtoetzsch, M.: Wikidata: A free collaborative knowledgebase. Communications of the ACM 57(10), 78–85 (2014)
[7] Auer, S., et al.: DBpedia: A nucleus for a web of open data. In: International Semantic Web Conference & Asian Semantic Web Conference, pp. 722–735 (2007)
[8] Singha, A.: Official Google blog: Introducing the knowledge graph: Things not strings. Available at: https://www.blog.google/products/search/introducing-knowledge-graph-things-not/. Accessed 16 January 2022
[9] Jernigan, D.B.: Update: Public health response to the coronavirus disease 2019 outbreak—United States, February 24, 2020. Morbidity & Mortality Weekly Report 69(8), 216–219 (2020)
[10] Quinlan, J.: C4.5: Programms for machine learning. Morgan Kaufmann Publishers, San Francisco (1995)
[11] Asahara, M., Matsumoto, Y.: Japanese named entity extraction with redundant morphological analysis. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 8–15 (2003)
[12] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back propagating errors. Nature 323, 533–536 (1986)
[13] Lacroix, T., Usunier, N., Obozinski, G.: Canonical tensor decomposition for knowledge base completion. In: Proceedings of the 35th International Conference on Machine Learning, pp. 2863–2872 (2018)
[14] Balažević, I., Allen, C., Hospedales, T.M.: TuckER: Tensor factorization for knowledge graph completion. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5185–

5194 (2019)
[15] Yang, B., et al.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2015)
[16] Trouillon, T., et al.: Complex embeddings for simple link prediction. arXiv preprint arXiv:1606.06357 (2016)
[17] Hitchcock, F.L.: The expression of a tensor or a polyadic as a sum of products. Journal of Mathematical Physics 6(1), 164–189 (1927)
[18] Zhang, C., et al.: Theoretical knowledge graph reasoning via ending anchored rules. arXiv preprint arXiv:2011.06174 (2020)
[19] Bordes, A., et al.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2 (NIPS' 13), pp. 2787–2795 (2013)
[20] Sun, Z., et al.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
[21] Qu, M., et al.: Rnnlogic: Learning logic rules for reasoning on knowledge graphs. arXiv preprint arXiv:2010.04029 (2020)
[22] Das, R., et al.: Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. arXiv preprint arXiv:1711.05851 (2017)
[23] Nguyen, D.Q., et al.: STransE: A novel embedding model of entities and relationships in knowledge bases. In: Proceedings of NAACL-HLT 2016, pp. 460–466 (2016)
[24] Dettmers, T., et al.: Convolutional 2d knowledge graph embeddings. arXiv preprint arXiv: 1707.01476 (2017)
[25] Schlichtkrull, M., et al.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference (ESWC), pp. 593–607 (2018)
[26] Guo, L., Sun, Z., Hu, W.: Learning to exploit long-term relational dependencies in knowledge graphs. In: International Conference on Machine Learning (ICML), pp. 2505–2514 (2019)
[27] Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: International Conference on Machine Learning, pp. 809–816 (2011)
[28] Avidan, S.: Ensemble tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(2), 261–271 (2007)
[29] Araújo, M.B., New, M.: Ensemble forecasting of species distributions. Trends in Ecology & Evolution 22(1), 42–47 (2007)
[30] Zhou, X., Liu, J., Xiong, J.: Predicting distant metastasis in breast cancer using ensemble classifier based on context-specific mirna regulation modules. In: 2012 IEEE International Conference on Bioinformatics and Biomedicine, pp. 1–6 (2012)
[31] Balazevic, L., Allen, C., Hospedales, T.M.: Hypernetwork knowledge graph embeddings. arXiv preprint arXiv: 1808.07018 (2018)
[32] Zhang, W., et al.: Interaction embeddings for prediction and explanation in knowledge graphs. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM ' 19), pp. 96–104 (2019)

[33] Liu, H., Wu, Y., Yang, Y.: Analogical inference for multi-relational embeddings. In: Proceedings of the 34th International Conference on Machine Learning (ICML' 17), pp. 2168–2178 (2017)
[34] Guo, S., et al.: Jointly embedding knowledge graphs and logical rules. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), pp. 192–202 (2016)
[35] Qu, M., et al.: Rnnlogic: Learning logic rules for reasoning on knowledge graphs. arXiv preprint arXiv:2010.04029 (2020)

Author Biography

Ting Jia received her undergraduate degree in Computer Science from Beijing Jiaotong University in 2019. She is currently a graduate student at the School of Computer and Information Technology, Beijing Jiaotong University. Her research interest is knowledge graph reasoning and medical knowledge engineering. ORCID: 0000-0003-2830-5626

Yuxia Yang received her Master' s degree in Computer Technology from Beijing Jiaotong University in 2021. During her studies, her research interest was learning to rank, with main research directions in drug classification and drug redirection. She is currently an engineer at the Railway Group Information Technology Institute. ORCID: 0000-0003-2035-9902

Xi Lu received his Master' s degree in Computer Technology from Beijing Jiaotong University in 2021. During his studies, his research interest was data mining, with main research direction in medical natural language text processing. He has conducted several experiments in medical text processing and co-published papers. He is currently a software development engineer at JD. ORCID: 0000-0002-0387-5924

Qiang Zhu received her Ph.D. degree in Biomedical Engineering from Tianjin University in 2002. She is currently an Associate Professor with the School of Computer and Information Technology at Beijing Jiaotong University. Her research interests cover medical artificial intelligence, medical information processing, big data, and data mining. ORCID: 0000-0002-4677-1244

Kuo Yang received his Ph.D. degree in Computer Science from Beijing Jiaotong University in 2020. He is currently a lecturer with the School of Computer and Information Technology at Beijing Jiaotong University. His research interests cover medical data mining, network medicine, bioinformatics, and artificial intelligence in medicine. He has published over 20 peer-reviewed papers in journals including *Nucleic Acids Research* (NAR), *Journal of the American Medical Informatics Association* (JAMIA), and *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (IEEE TCBB). ORCID: 0000-0003-0736-4512

Xuezhong Zhou received his Ph.D. degree in Computer Science from Zhejiang University in 2005. He is currently a Professor with the School of Computer and Information Technology at Beijing Jiaotong University. His research interests

cover medical data mining, network medicine, clinical decision support, and medical knowledge engineering. He has published over 100 peer-reviewed papers in journals including *Nature Communications*, *Ebiomedicine*, and *Nucleic Acid Research* (NCR). ORCID: 0000-0002-4713-3594

Note: Figure translations are in progress. See original paper for figures.