Rule-Aware Reinforcement Learning for Knowledge Graph Reasoning

Zhongni Hou[1,2], Xiaolong Jin[1,2], Zixuan Li[1,2], Long Bai[1,2]
[1]School of Computer Science and Technology, University of Chinese Academy of Sciences;
[2]CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences
{houzhongni18z,jinxiaolong,lizixuan,bailong18b}@ict.ac.cn

Abstract

Multi-hop reasoning is an effective and explainable approach to predicting missing facts in Knowledge Graphs (KGs). It usually adopts the Reinforcement Learning (RL) framework and searches over the KG to find an evidential path. However, due to the large exploration space, the RL-based model struggles with the serious sparse reward problem and needs to make a lot of trials. Moreover, its exploration can be biased towards spurious paths that coincidentally lead to correct answers. To solve both problems, we propose a simple but effective RL-based method called RARL (Rule-Aware RL). It injects high quality symbolic rules into the model's reasoning process and employs partially random beam search, which can not only increase the probability of paths getting rewards, but also alleviate the impact of spurious paths. Experimental results show that it outperforms existing multi-hop methods in terms of Hit@1 and MRR.

1 Introduction

Knowledge Graphs (KGs), which store facts as triples in the form of (*subject entity*, *relation*, *object entity*), benefit various NLP applications (Lan and Jiang, 2020; Wang et al., 2019b; He et al., 2017). However, existing KGs face with serious incompleteness despite of their large scales. Therefore, KG completion, which aims to reason missing facts based on existing triples, has been an important research area.

The past decade has witnessed the rise of embedding-based reasoning methods on KGs (Bordes et al., 2013; Yang et al., 2014; Balažević et al., 2019). However, due to their black-box nature, these methods cannot provide interpretations for a specific prediction (Ji et al., 2020; Sadeghian et al., 2019). Recently, there has been growing interest in using multi-hop reasoning to improve the interpretability (Gardner et al., 2013; Rocktäschel and Riedel, 2017). This approach usually adopts Reinforcement Learning (RL) to find a reasoning path (Xiong et al., 2017; Das et al., 2018; Hildebrandt et al., 2020). Starting from the query entity, the RL-based model sequentially selects an outgoing edge and transits to a new entity until it arrives at the target.

However, due to the complexity of the KG, the number of paths grows exponentially when the reasoning hop increases. Most of paths cannot arrive at correct answers, and cannot receive a none-zero reward, which is also called the "sparse reward problem" (Nair et al., 2018). Moreover, since golden paths are not available in the training process, the RL-based model may coincidentally reach the target via a meaningless path (i.e. spurious paths). Take the query (*Captain America, director, ?*) as an instance. Although the path (*Captain America, country, US, lives in*$^{-1}$*, Peter Farrelly*), can arrive at the target. It is semantically inconsistent with the query relation *director* and is an accidental success. One trouble is that the RL-based model relies heavily on rewards and reinforces the past actions receiving high rewards regardless of their path quality. In addition, in large scale KGs, there are more spurious paths than correct ones (Lin et al., 2018). It is more easier for the model to discover spurious ones first other than the true and meaningful ones. If the model finds spurious ones first, these spurious paths will lead to a biased exploration and induce negative influence to the reasoning process (Guu et al., 2017; Lin et al., 2018).

Lin et al. (2018) uses shaped rewards calculated by pre-trained embedding-based models and an action dropout mechanism to solve the above two challenges, respectively. However, its performance largely depends on the embedding-based model used. In addition, embedding-based model increases the opacity of the reasoning process. Motivated by this, we focus on the action selection

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4687–4692

strategy and propose RARL (Rule Aware RL), a simple but effective model to solve the above two challenges. RARL introduces high quality rules as prior information about actions and explores K paths in one episode. It selects actions from three parts: actions matching rules, actions with high scores, and actions randomly sampled. The former two parts can increase the probability of reasoning paths arriving at targets. The later one allows the model to explore a more diverse path set and thus avoids the model adhering to the past actions receiving high rewards, which can naturally mitigate the impact of spurious paths.

We evaluate RARL on three benchmark datasets, and experimental results show the effectiveness of RARL when compared with existing multi-hop methods.

2 Preliminaries

Let $\mathcal{E}$ be the set of entities and $\mathcal{R}$ be the set of relations, a knowledge graph can be represented as $\mathcal{G} = \{(e_s, r, e_t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. In this paper, we focus on the standard link prediction task. Given a query of the form $(e_s, r_q, ?)$, the reasoning model is expected to predict the correct answer e_t after traversing over the graph.

2.1 The RL-based Knowledge Reasoning Framework

Following (Das et al., 2018), when given a query $(e_s, r_q, ?)$, the RL-based model can be viewed as an agent, which interacts with the KG environment and aims to find a reasoning path $p = (e_s, r_1, e_1, ...)$ to explicitly show how to conduct reasoning. The parameters of the policy defines a policy. At each time t, the agent selects an action a_t, i.e. an outgoing edge of the current position e_t, to expand the path using a policy. Here, we define $A_t = \{(r', e')|(e_t, r', e') \in \mathcal{G}\}$ as the possible actions at time t. The model first uses a Long Short-Term Memory network (LSTM) to encode the path history into a vector $\mathbf{h}_t$. Then, the policy network π_θ (a two-layer feed-forward network) calculates a distribution over all possible actions in A_t.

$$\pi_\theta(a_t|e_t) = \sigma(\mathbf{A}_t(W_2 ReLU(W_1[\mathbf{h}_t; \mathbf{e}_t; \mathbf{r_q}]))), \quad (1)$$

where $\mathbf{e}_t \in \mathbb{R}^d$ and $\mathbf{r_q} \in \mathbb{R}^d$ are embeddings of e_t and r_q, respectively. $\mathbf{A}_t \in \mathbb{R}^{|A_t| \times 2d}$ is the stack of all actions embeddings in A_t and σ denotes the softmax operator. After this, the next edge is selected via an ϵ-greedy action selection strategy.

A binary reward $R(p)$ is observed after the maximum time step T: $R(p) = 1$ if the path ends at the correct answer and 0 otherwise.

The objective of the model is to maximize the expected reward:

$$J(\theta) = \sum_{(e_s, r_q, e_t) \in \mathcal{G}} \sum_{\hat{z} \in P(e_s, r_q)} R(\hat{z}) \pi_\theta(\hat{z}|e_s, r_q), \quad (2)$$

where $P(e_s, r_q)$ is the set of all reasoning paths related to the given query $(e_s, r_q, ?)$. The optimization is then performed by using REINFORCE algorithm (Williams, 1992).

2.2 Beam Search

In the RL context, Beam Search (BS) (Sutskever et al., 2014) stores top-K scoring partially constructed paths at each time step, where K is known as the beam size. At each time t, BS extends itself via the following process. Let us denote the paths set held by BS at the end of time t as $B_t = \{p_{[1:t]}, ..., p_{[K:t]}\}$. For each path $p = (e_s, r_1, ..., e_t) \in B_t$, we first generate its candidate paths $cand(p)$,

$$\begin{aligned} cand(p) &= cond(e_s, r_1, ..., e_t) \\ &= \{(e_s, r_1, ..., e_t, r', e')|(e_t, r', e') \in \mathcal{G}\}. \end{aligned} \quad (3)$$

Each candidate path $p \in cand(p)$ is associated with a score $s(p)$ calculated by the policy network. Here, $s(p) = \pi_\theta((r', e')|e_t)$. Further, we take the union of these candidate paths $\mathcal{B}_t = \bigcup_{p \in B_t} cand(p)$. A new beam B_{t+1} is generated by picking the K top-most elements in $\mathcal{B}_t$.

3 The RARL Model

As illustrated in Figure 1, the RARL model consists of two parts: the KG environment and the agent. By interacting with the environment, the agent employs a beam search based action selection strategy and picks K actions to extend the beam in one episode. The action selection strategy selects actions from three parts: actions matching rules, actions with high scores, and actions randomly sampled. After the maximum time step, the agent will receive binary rewards.

3.1 Rule Based Action Selection

In a typical KG, when the path length increases, finding a non-zero reward is exponentially more difficult. Learning from such sparse rewards requires lots of effective exploration. However, in the beginning, due to the randomly initialized parameters,

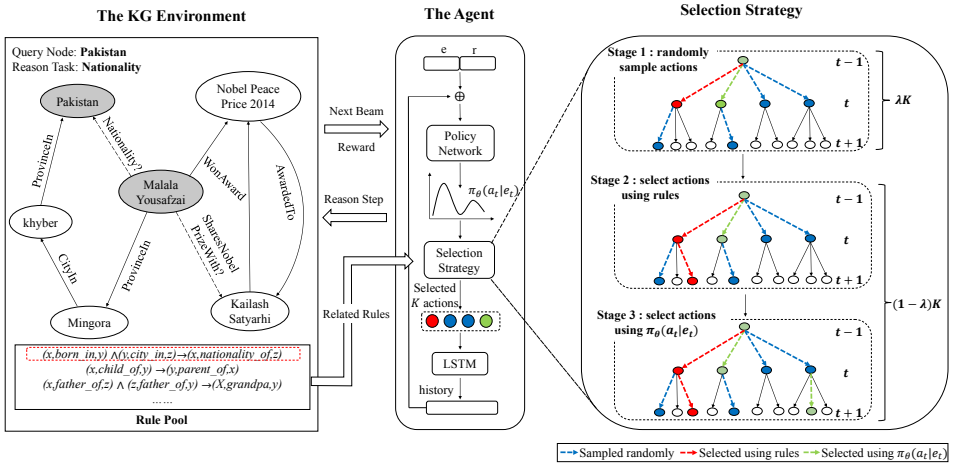

Figure 1: The reasoning process of RARL. Left: The KG environment. The dashed edges are part of queries and the solid edges are observed facts. Middle: The structure of the agent. By interacting with the environment, it picks K actions to extend the beam in one episode. Right: An illustration of the action selection strategy at time t. It first randomly samples λK actions and then selects the remaining $(1-\lambda)K$ actions using rules and probabilities.

the model chooses actions randomly and can hardly arrive at targets. This makes the sparse reward problem even worse (Xiong et al., 2017; Hare, 2019). Considering that rules precisely characterize a mapping from query relations to semantic composition paths (Zhang et al., 2019), RARL utilizes rules as prior information about actions to increase the probability of paths receiving rewards, which can also help to facilitate effective exploration.

The rules mined from KGs are in the form of $head \leftarrow body$, where the head is an atom $r(a, b)$ and the body is in the format of: $r(x_0, x_1) \wedge \ldots \wedge r(x_n, x_{n+1})$. Note that $r(x_i, x_j)$ is equivalent to the fact triple (x_i, r, x_j).

Given the query relation r_q, RARL first selects rules R_{r_q} whose heads are identical to r_q from the rule pool. At each time step t, it maintains a beam B_t of K paths. For each path $p = \{e_s, r_1, e_1, ..., r_t, e_t\} \in B_t$, RARL expands its candidate paths based on the outgoing edges of e_t. Next, for all candidate paths in $\mathcal{B}_t$, only those in which relation sequence can match related rules from left to right are selected. For instance, suppose R_{r_q} contains only one rule $r_q \leftarrow r_1 \wedge r_2$, given two candidate paths $(e_s, r_1, e_1, r_2, e_2), (e_s, r_2, e_3, r_3, e_4)$, only the relation sequence in the former path can successfully matches the rule. As a result, the former one will be selected to generate B_{t+1}. If the number of candidate paths matching rules excesses the beam size, RARL selects top-K paths from these paths matching rules according to their scores calculated by the policy network. If not, remaining paths not matching rules are selected as a compliment. To make a balance between actions generated by free exploration and actions matching rules, RARL randomly masks some related rules to shrink the number of paths matching rules.

3.2 Partially Random Beam Search

To ease the impact of spurious paths, we try to prevent the RL-based model from the obsession of spurious paths and induce diversity during BS. Inspired by (Guu et al., 2017), we introduces partially randomness to standard BS, to fight against the impact of spurious paths.

Like regular beam search, at time t, RARL computes the set of all candidate paths $\mathcal{B}_t$ and sorts them by their scores computed by the policy network π_θ. Instead of selecting K highest-scoring candidate paths, RARL randomly chooses λK candidate paths from $\mathcal{B}_t$ and remaining paths are chosen according to their scores. In this way, low-scoring paths discarded in standard BS can also have the chance to be explored. Besides, the randomness can avoid the model sticking to the paths getting rewards. In the experiment, RARL selects paths with replacement when available actions smaller than λK.

3.3 The Overall Selection Strategy

RARL selects candidate actions by three stages: (1) Randomly sample λK actions based on the current position; (2) Select actions matching the

	Model	UMLS			WN18RR			FB15K-237		
		Hit@1	Hits@10	MRR	Hit@1	Hits@10	MRR	Hit@1	Hits@10	MRR
Multi Hop	NerualLP(Yang et al., 2017)	.643	.962	.778	.376	**.657**	.463	-	-	-
	MINERVA(Das et al., 2018)	.728	.968	.825	.413	.513	.406	.405	.583	.468
	AnyBURL(Wang et al., 2019a)	.657	.966	-	.431	.526	-	.220	.335	-
	Coper-MINERVA(Stoica et al., 2020)	.778	**.974**	.854	.426	.510	.465	.484	.630	.536
	RARL (ours)	**.803**	.970	**.866**	**.442**	.533	**.469**	**.516**	**.634**	**.557**
Embedding	DistMult(Yang et al., 2014)	.821	.967	.868	.431	.524	.462	.477	.642	.535
	ComplEx(Trouillon et al., 2016)	.890	.992	.934	.418	.480	.437	.496	.687	.563
	ConvE (Dettmers et al., 2017)	.932	.994	.957	.403	.540	.449	.480	.663	.544
	TuckER (Balažević et al., 2019)	.822	.997	.907	.443	.526	.470	-	-	-

Table 1: Link prediction results on UMLS, WN18RR, and FB15K-237. Best scores among the multi-hop methods and embedding-based methods are bold and underlined, respectively.

related rules based on the history; (3) Select actions in descending order of scores. If the number of actions matching rules from the second stage excesses $(1-\lambda)K$, then RARL selects the top $(1-\lambda)K$ actions according to their scores. If not, it continue to select actions via the third stage. The total size of actions selected from the last two parts are $(1-\lambda)K$.

4 Experiments

4.1 Experimental Setup

Datasets and Rules We adopt three datasets to evaluate the performance of RARL for link prediction: UMLS (Kok and Domingos, 2007), WN18RR (Dettmers et al., 2017), and FB15K-237 (Toutanova et al., 2015). For FB15k-237, 20 relations in the film field are selected. Following (Niu et al., 2020), We use AIME+ (Galárraga et al., 2015) to automatically extract rules, and we limit the maximum length of rules to 2.

Table 2 lists the statistics of rules with various confidence thresholds mined from these three datasets.

Model	Various Confidence Thresholds				
	0.5	0.6	0.7	0.8	0.9
UMLS	2,154	1,678	1,159	561	170
WN18RR	5	4	3	3	3
FB15K-237	2,044	1,621	1,255	912	565

Table 2: Statistics of rules mined on the three datasets.

Hyperparameters We set the dimensions of entity and relation embeddings within (50, 200). A three-layer LSTM is used as the path encoder and its hidden dimension is set in (100, 200). The λ is set as 0.9, 0.4 and 0.7 for UMLS, WN18RR and FB15K-237, respectively, according to the average degree of nodes and the average number of relation rules on each dataset.

4.2 Link Prediction Results

Table 1 summarizes the experimental results of our proposed approach and the baselines. As shown in Table 1, RARL achieves competitive results over multi-hop reasoning methods. On FB15K-237, RARL outperforms all baselines in terms of Hit@1, Hits@10, and MRR. On WN18RR and UMLS, the RARL achieves the best results in terms of Hit@1 and MRR. The Hit@1 results emphasize the superiority of our approach in high-precision link prediction and confirm the effectiveness of high quality rules. We also notice that the emebedding-based methods perform better on UMLS and FB15K-237 compared with multi-hop reasoning methods. One reason is that the multi-hop reasoning methods are more sensitive to the sparsity and incompleteness of graphs compared with embedding-based methods. It is hard for them to find evidential paths reaching targets via strictly searching in the KG. While the embedding-based methods(Lin et al., 2018; Fu et al., 2019) map entities and relations into a unified semantic space to capture inner connections, which relaxes this restriction.

4.3 Ablation Study

We perform an ablation study to look deep into the framework of RARL. We deactivate the validity of rule information, random mechanism from RARL. The MRR results are summarized in Table 3. It can be observed that removing each reasoning component of RARL results in a significant performance drop on UMLS and WN18RR. On FB15K-237, removing rule information seems like to be no influence. As a result, we further conduct an analysis experiment using w/o rule setting and found lower results on the testing set. This performance gap may be caused by the difference of data distribution between the testing set and the validation set.

Besides, our ablation study shows that removing partial randomness has a greater negative impact on reasoning performance. This suggests that increasing the exploration diversity to get more valid path patterns is important in training stage.

	UMLS	WN18RR	FB15K-237
RARL w/o Rule	.813	.448	.551
RARL w/o Random	.792	.438	.501
RARL	.872	.455	.551

Table 3: Ablation study of the proposed method.

5 Conclusions

In this paper, we introduced RARL, a new RL-based method for knowledge graph reasoning. RARL makes use of high-quality symbolic rules and partical random beam search jointly and efficiently fights against the sparse reward and spurious path problems. Experimental results demonstrate that RARL achieves better performance compared with existing multi-hop methods in terms of both Hit@1 and MRR.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under grant 2016YFB1000902, the National Natural Science Foundation of China under grants U1911401, 61772501, 62002341 and U1836206, and the GFKJ Innovation Program.

References

Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2017. Convolutional 2d knowledge graph embeddings. *arXiv preprint arXiv:1707.01476*.

Cong Fu, Tong Chen, Meng Qu, Woojeong Jin, and Xiang Ren. 2019. Collaborative policy learning for open knowledge graph reasoning. *arXiv preprint arXiv:1909.00230*.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal—The International Journal on Very Large Data Bases*, 24(6):707–730.

Matt Gardner, Partha Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 833–838.

Kelvin Guu, Panupong Pasupat, Evan Zheran Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. *arXiv preprint arXiv:1704.07926*.

Joshua Hare. 2019. Dealing with sparse rewards in reinforcement learning. *arXiv preprint arXiv:1910.09281*.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. *arXiv preprint arXiv:1704.07130*.

Marcel Hildebrandt, Jorge Andres Quintero Serna, Yunpu Ma, Martin Ringsquandl, Mitchell Joblin, and Volker Tresp. 2020. Reasoning on knowledge graphs with debate dynamics. *arXiv preprint arXiv:2001.00461*.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*.

Stanley Kok and Pedro Domingos. 2007. Statistical predicate invention. In *Proceedings of the 24th international conference on Machine learning*, pages 433–440.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. Association for Computational Linguistics.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. *arXiv preprint arXiv:1808.10568*.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299. IEEE.

Guanglin Niu, Yongfei Zhang, Bo Li, Peng Cui, Si Liu, Jingyang Li, and Xiaowei Zhang. 2020. Rule-guided compositional representation learning on knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2950–2958.

Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems*, pages 3788–3800.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. Drum: End-to-end differentiable rule mining on knowledge graphs. In *Advances in Neural Information Processing Systems*, pages 15347–15357.

George Stoica, Otilia Stretcu, Emmanouil Antonios Platanios, Tom Mitchell, and Barnabás Póczos. 2020. Contextual parameter generation for knowledge graph link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3000–3008.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27:3104–3112.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1499–1509.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML).

Po-Wei Wang, Daria Stepanova, Csaba Domokos, and J Zico Kolter. 2019a. Differentiable learning of numerical rules in knowledge graphs. In *International Conference on Learning Representations*.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019b. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv preprint arXiv:1707.06690*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems*, pages 2319–2328.

Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Iteratively learning embeddings and rules for knowledge graph reasoning. In *The World Wide Web Conference*, pages 2366–2377.