Split Multi-Hop Knowledge Graph Reasoning with Reward Shaping

Blaine Hill
Department of Computer Science
University of Illinois Urbana-Champaign
Champaign, IL
blaineh2@illinois.edu

Yingzhuo Yu
Department of Computer Science
University of Illinois Urbana-Champaign
Champaign, IL
yy56@illinois.edu

1 Introduction

Knowledge Graphs (KGs) are relational representations that contain nodes (denoting a subject or entity) and edges (denoting a verb or dependence) which connect nodes in either a uni-directional or bi-directional manner. Specifically, a KG consists of many factual triplets of the form $\langle h, r, t \rangle$ representing a head node, relation edge, and tail node. Through this simple model, mathematical symmetry/asymmetry, inversion, and composition can be represented and used in many applications. Knowledge Graph Reasoning (KG Reasoning), as seen as in Figure 1, aims to solve different logical reasoning tasks, whereby inferring new knowledge from existing ones [8]. For example, the existing path information in the KG can be used to predict missing links [1, 14, 12]; and the KG structure information can be used to answer complex questions [11, 9].

One area of KG Reasoning research formulates the problem statement as sequential decision making solved via reinforcement learning (RL) [13]. MINERVA (Das et al., 2018) employs the REINFORCE algorithm (Williams, 1992) to develop an end-to-end model that can perform multi-hop KG query answering [2, 15]. The trained agent searches through the KG starting from the given source entity, for the candidate answers related to the query relation. Notably, the agent does not rely on any pre-computed paths to arrive at the answers. This method is referred to as *walk-based query-answering* (QA). However, this approach poses challenges for training as practical KGs are often incomplete, leading to false negatives and false positives in the returns from the trajectories. Lastly, REINFORCE is an on-policy RL algorithm that can bias the policy towards spurious paths found early in training.

[7] proposes two improvements to reinforcement learning in the walk-based QA framework. Firstly, instead of using a binary reward, they use pre-trained embedding-based models to estimate a soft reward for target entities, known as *reward shaping*. Secondly, they perform *action dropout* to enforce effective exploration and dilute the negative impact of spurious ones. We will focus on the former of the two methods as it can be improved via transfer learning.

To this end, we extend [7] in the area of transfer learning on the UMLS benchmark dataset: we will introduce how we split a dataset into rich and sparse KGs, explore different pretrainings of a "Reward Shaper" module that controls the mechanism of reward shaping using these KGs, and compare them to [7] as a baseline. Additionally, we explore the using pretrained BERT[4] embeddings and various Prompt Learning techniques to improve reward shaping.

2 Related Work

2.1 KG Reasoning as Reinforcement Learning

A knowledge graph is represented formally as $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ denotes the entities set and $\mathcal{R}$ denotes the relations set. Each link in the graph is directed and can be represented as $l = (e_s, r, e_o) \in \mathcal{G}$, indicating a fact or triple. Given a query $(e_s, r_q, ?)$ where e_s denotes the source entity and r_q

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

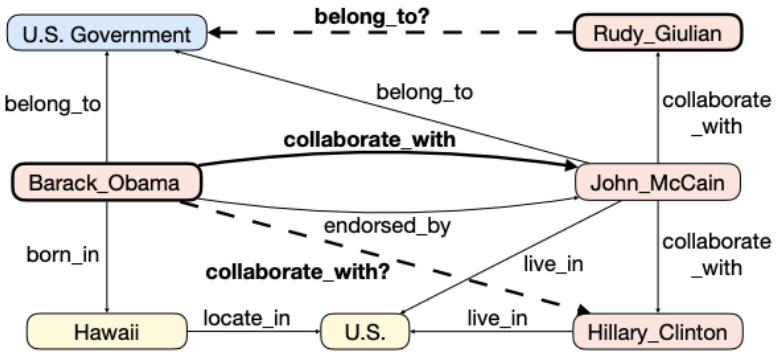

Figure 1: An incomplete KG where there are some connections (represented by dashed lines) that are missing and could potentially be deduced from the information already present (represented by solid lines).

denotes the relation of interest, the aim of KG Reasoning is to efficiently search $\mathcal{G}$ and gather the set of possible answers $E_o = e_o$ such that $(e_s, r_q, e_o) \in \mathcal{G}$. From this set, usually the softmax is taken and the top answer returned as the prediction.

This algorithm is representable as a Markov Decision Process (MDP) [13]: starting from e_s, the agent sequentially selects an outgoing edge l and deterministically transitions to a new entity in connection with l until it selects a terminal action. The MDP is comprised of the following components [2]:

States. Each state $s_t = (e_t, (e_s, r_q)) \in \mathcal{S}$ is a tuple where e_t is the entity visited at step t and (e_s, r_q) are the source entity and query relation. e_t can be viewed as state-dependent information while (e_s, r_q) are the global context shared by all states [7]. This is an important distinction because the agent will ideally learn the rich relationships between various e_t while conditioning on (e_s, r_q).

Actions. At time t, the set of possible actions, $A_t \in \mathcal{A}$, is $\{(r', e') \mid (e_t, r', e') \in \mathcal{G}\}$ (the collection of outgoing edges belonging to e_t). To introduce the agent to a terminating action, a self-loop edge is added to every $A_t \in \mathcal{A}$.

Transition. A transition function $\delta : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is defined by $\delta(s_t, A_t) = \delta(e_t, (e_s, r_q), A_t)$.

Rewards. The RL agent will obtain a terminal reward of 1 if it arrives at a correct target entity and 0 otherwise. As we will see, this is a brittle approach and does not allow for policies to learn when they achieve close but incorrect predictions.

$$R_b(s_T) = \mathbf{1}\{(e_s, r_q, e_T) \in \mathcal{G}\}$$

2.2 Reward Shaping

The given reward function, provides a binary reward based solely on the observed answers in $\mathcal{G}$, which is incomplete. False negative search results receive the same reward as true negatives, which can be problematic. To address this issue, the authors propose a reward shaping strategy using existing KG embedding models designed for KG completion.

These embedding models map entities $\mathcal{E}$ and relations $\mathcal{R}$ to a vector space and estimate the likelihood of each fact $l = (e_s, r, e_t) \in \mathcal{G}$ using a composition function of the entity and relation embeddings $f(e_s, r, e_t)$. This function f is trained by maximizing the likelihood of all facts in $\mathcal{G}$.

The proposed reward shaping strategy involves adding a soft reward to the binary reward. If the destination entity e_T is a correct answer according to $\mathcal{G}$, the agent receives a reward of 1. Otherwise, the agent receives a fact score estimated by $f(e_s, r_q, e_T)$. The shaping of the reward is controlled by a "Reward Shaper" module, which adds the soft reward to the binary reward.

By using this Reward Shaper, the agent can receive a higher reward for finding correct answers and a lower reward for false negatives. The authors note that the embedding model used in this strategy can be replaced by any state-of-the-art model, providing flexibility in the choice of model.

We refer the reader to [7] for further information on policy search, policy optimization, and action dropout background.

2.3 Pre-trained Language Model

In recent years, the use of large pre-trained language models, such as BERT (Bidirectional Encoder Representations from Transformers), has shown great promise in various natural language processing (NLP) tasks. Consisting of multiple layers of encoder blocks, it allows the model to process both left and right context during pretraining, enabling it to capture complex patterns and rich contextual relationships within language. Fine-tuning BERT[4] involves taking the pretrained model and fine-tuning it on a specific downstream task. This approach has been shown to improve the performance of the model on the target task with relatively small amounts of labeled data.

In the context of a KG embedding, fine-tuning BERT has been used to encode textual information associated with entities and relationships in the graph into a low-dimensional vector representation. KG-BERT [17] has demonstrated that fine-tuning BERT for triple classification in a KG can lead to significant improvements in performance compared to other embedding methods. In this study, we plan to add contextual embedding in pretraining Reward Shaping module by using the name of the head entities and the relations in the KG as input and fine-tuning BERT to assign score for each tail entity.

2.4 Prompt Learning

Prompt learning is a machine learning approach that involves using natural language prompts to guide the model's learning process. In prompt learning, a prompt is a specific input given to the pretrained model to help it understand what kind of output is desired. Previous works have shown its success in many NLP tasks, especially under low-data scenario. [5] applied cloze-style language prompts to pre-trained language models on fine-grained entity typing. By developing the prompt learning pipeline consisting of template, entity-oriented verbalizers and masked language modeling, it demonstrates that prompt-learning methods outperform the normal fine-tuning baselines. OpenPrompt[6] established a unified, extensible and easy-to-use framework to conduct prompt-learning over various pretrained language models, which facilitates researchers to efficiently deploy prompt-learning pipelines. In this project, we utilized the OpenPrompt framework for building the prompt-learning pipeline.

3 Problem Statement

In the original work [7], the pretraining of the Reward Shaper is mainly based on three path-based knowledge graph embedding algorithms: ConvE[3], ComplEx[14] and DistMult[16]. These three embedding methods aim to represent entities and relationships in a low-dimensional vector space by using different approaches to capture the spatial relationship of entities in the KG.

In this project, we attempt to improve the contextual embeddings of entities and relationships in reward shaping: by pretraining the Reward Shaper a rich KG (simulated by the original KG), we imitate the real-world scenario of a large, accessible general KG that envelops a sparse KG of true interest. Our hypothesis is that adding contextual information to this rich, general KG can help improve generalization of the reward shaping, which in turn will assist in guiding the policy as it achieves better predictions.

4 Proposed Method

4.1 Transfer Learning with Rich and Sparse KGs

One plausible challenge of KG Reasoning in practice is where a large, well-connected general KG is available and one wishes to instead learn a policy over a small, relatively tractable KG for which there is limited sampling. A clever way to circumvent this issue is to learn general knowledge from the rich KG and fine-tune on the sparse KG.

We simulate these KGs by masking. In our experiments we simply treat a source KG dataset as our "rich" KG and we mask 50% of the nodes and edges to obtain a "sparse" KG, dubbed "Split

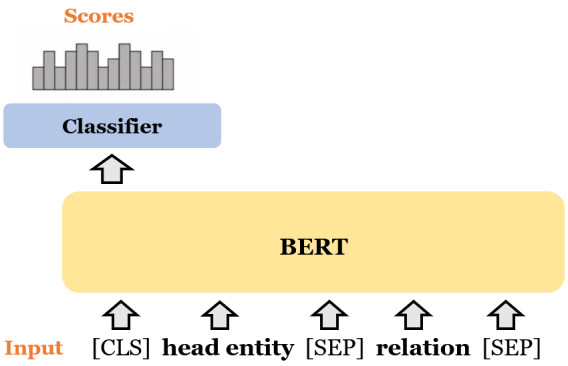

Figure 2: Model architecture of reward shaping with BERT contextualization

Multi-Hop KG Reasoning". Through this, we simulate a well-trained Reward Shaper on some incomplete KG.

4.2 Reward Shaping with BERT Contextualization

To get reward scores for each tail entity given a head entity and relation, the first approach is to formulate it as a multi-label classification problem. The model architecture is shown in Figure 4.2. We fine-tune the BERT[4] with a classifier. We treat each tail entity as independent label and apply the Sigmoid function to predict the score for each tail entity, given the head entity and relation. In the supervised training process, if there is a relationship between head entity and tail entity, the ground truth label for that tail entity is one. Otherwise, the score is zero. However, in the actual training, we applied the same smoothing approach to the ground truth label as [7] and took Binary Cross Entropy as the loss function to fine-tune the whole model system.

4.3 Reward Shaping with Prompt Learning

Another method to pretrain the Reward Shaper is based on Prompt Learning, as shown in Figure 4.3. We first set the template to guide the language model to build the relationship between the head entity and tail entity in the training KG. The template is set as $[head\ entity]\ is\ related\ to\ [tail\ entity]\ through\ relationship\ of\ [relation]$. Then, while still in the training process and given the head entity and relation, we fill out the corresponding token and get the prompt ready for the pretrained language model. As an aside, we selected the T5 model for prompting [10] due to its impressive performance on a variety of NLP tasks, as well as its ability to generate high-quality natural language text. Based on T5, we can get the probability of the masked token, namely the tail entity token and use its as the score. The training process and loss function remain the same as the previous method of BERT Contextualization.

we get the score assigned to the tail entity based on the probability of tokens from pre-trained language model (T5, BERT)

5 Experiments

5.1 Evaluation

To evaluate both reward shaping and Reinforcement Learning training, we transform each triple in the test set into a query head entity and a query relation. Specifically, given the head entity and a relation, the models generate a list of candidate tail entities ranked by confidence scores. And then, we calculate two types of ranking-based performance metrics for evaluation. Hits@k measures the percentage of examples where the correct answer is ranked within the top-k positions of the ranked list. Mean reciprocal rank (MRR) is calculated by taking the reciprocal of the rank of the first correct entity in the ranked list, and then averaging the value across all queries.

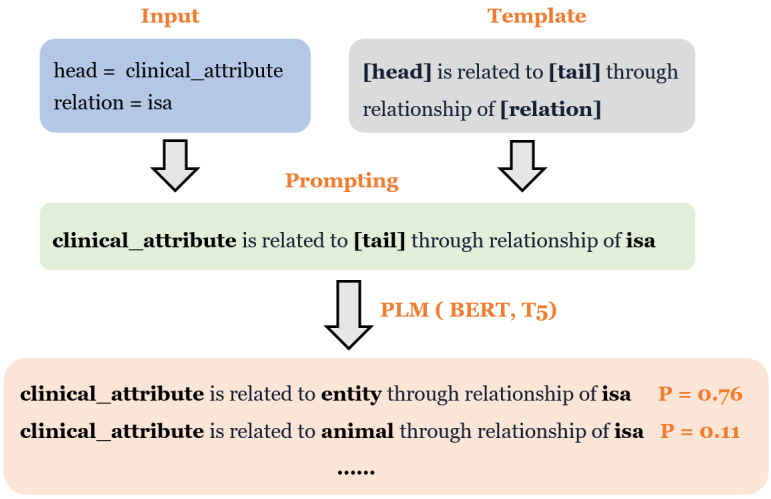

Figure 3: Overall pipeline of reward shaping with prompt learning

5.2 Results from UMLS, Kinship, and FB15k-237 using 50% Masking Sparse KG

First, we split the UMLS dataset into two KGs using Algorithm 1:

Algorithm 1: Knowledge Graph Masking Algorithm

Input: input complete KG: *data file*, sparsity_nodes: *float*, sparsity_edges: *float*
Output: rich_KG: *data file*, sparse_KG: *data file*
Mask Function:
    nodes_to_mask: *int* ← roundUp(sparsity_nodes × getNumberOfNodes(KG))
    randomly_mask(nodes_to_mask, KG)
    edges_to_mask ← round(sparsity_edges × number_of_edges(KG))
    unique_edges ← get_unique_edges(KG)
    mask_proportional_edges(edges_to_mask, unique_edges)
return

Algorithm 1 is designed such that maximal unique edges are preserved while maintaining randomness. This is done by randomly masking nodes, but proportionally reducing the number of edges (while still randomly selecting edges).

We then run a standard policy gradient algorithm to learn a policy for KG Reasoning on these KGs as a baseline. Lastly, we apply our Reward Shaper given by ConvE, ComplEx, and DistMult, aggregate their performance, and supplement the policy gradient algorithm. The results of this experiment are given in Table 1. Note that Hits@1 , 3, etc represent how often the correct entity was in the model's top X predictions. Additionally, MRR refers to mean reciprocal rank, which is the mean of $\frac{1}{r_{e_o}}$ from the test set.

As we can see, reward shaping played a huge role in improving the agent's ability to overcome spurious paths. However, a surprise is that training on the rich KG did not supercede the sparse KG in terms of performance. This is likely due to the size of UMLS and how the rich Reward Shaper overfit. Despite being the amalgation of many healthcare and biohealth vocabulary knowledge graph, it is still tiny in comparison to other benchmark datasets such as FB15k-237 or Kinship. UMLS consists of triples formed by 135 different entities and 49 different relations, for a total of 6529 ground truth triples out of a possible 893,025. It is therefore no suprise that training the Reward Shaper embeddings on this KG overfit the training data.

Experiment Configuration	UMLS				
	Hits@1	Hits@3	Hits@5	Hits@10	MRR
Sparse KG Policy Gradient	0.649	0.852	0.893	0.935	0.76
Rich KG Policy Gradient	0.728	0.900	0.930	0.958	0.822
Sparse KG Policy Gradient + Rich Reward Shaping	0.888	0.955	0.974	**0.986**	0.926
Sparse KG Policy Gradient + Sparse Reward Shaping	**0.893**	**0.971**	**0.980**	**0.986**	**0.934**

Table 1: Query answering performance comparison on UMLS dataset of multi-hop reasoning on Rich/Sparse KG with reward shaping pre-trained on Rich/Sparse KG.

Model Configuration	UMLS				
	Hits@1	Hits@3	Hits@5	Hits@10	MRR
ConvE Reward Shaping trained on Rich KG (baseline)	**0.888**	0.955	0.974	0.986	0.926
BERT Contextualization RS trained on Rich KG (ours)	0.776	0.955	0.974	0.991	0.871
Prompt Learning based RS trained on Rich KG (ours)	0.850	**0.989**	**0.992**	**0.995**	**0.930**

Table 2: Query answering performance comparison of multi-hop reasoning with reward shaping pre-trained by different embedding methods on UMLS dataset.

5.3 Results from UMLS using BERT and Prompt Learning-based Reward Shaping

We conducted the experiment of pre-training reward shaping with different methods on rich knowledge graph and then training RL agent with reward shaping on sparse knowledge graph. Table 2 shows the evaluation performance of RL-based multi-hop reasoning with our proposed contextual embedding reward shaping. The reward shaping with traditional embedding ConvE is our baseline.

We find that prompt learning based reward shaping module that is pretrained on rich knowledge graph has the best generalization for RL agent multihop reasoning on sparse knowledge graph. Prompt learning based method outperforms ConvE based method on most of evaluation metrics. We expected that adding contextual meaning of entities and relations in the knowledge graph can help generalize the scoring well on the following RL training. However, the BERT-based method performed worse than both the prompt learning-based and ConvE-based methods, which was unexpected.

One potential reason for the lower performance of the BERT-based method could be due to the way it encodes knowledge graph information and is treated as multi-label classification task. It is possible that the context embeddings learned by BERT do not capture the relevant information needed for effective reward shaping in this domain. And directly treating the tail entity as the labels when predicting by the classifier didn't take the contextual meaning of the tail entity itself into consideration. On the contrary, in the prompt learning based method, we filled the tail entity into the template together with the query tail entity and relation as prompt, we utilized all the contextual information of entities and relation. That's the potential reason why prompt learning-based method is the most effective for reward shaping in this domain. Additionally, further experimentation and analysis could be done to determine how to improve the performance of the BERT-based method and to explore how to further improve the performance of the prompt-learning, especially on Hits@1 metrics.

6 Conclusions

In order to apply [7], reward shaping can only be done if you have a sufficiently large KG such that meaningful embeddings can be learned. Because this situation is often not the case, we introduce Split Multihop KG Reasoning, which extends [7] to a transfer learning domain by pretraining the Reward Shaper module on a rich KG while applying it to a sparse KG. Additionally, we add BERT pretraining and Prompt Learning to reward shaping to improve its performance by taking advantage of the natural language representation in a KG.

We tested these ideas on UMLS, or the Unified Medical Language System, which is a small but diverse KG and showed that reward shaping had a huge impact on the performance of the RL agent. Counter to our hypothesis, training the Reward Shaper on the rich KG performed well, but not as well as training it on the sparse Reward Shaper, indicating the rich Reward Shaper was overfitting.

On the other hand, BERT contextualization and Prompt Learning performed much better than adjusting the Reward Shaper. Among those two approaches, prompt learning based reward shaping improves over baseline multi-hop reasoning RL models on most evaluation metrics. A further analysis shows that it's significant to apply the appropriate way to adding contextual information of knowledge graph into training reward shaping by prompt learning.

7 Future Work

Going forward, more replication experiments should be conducted to verify our findings. Additionally, locating "naturally occurring" KGs with similar state-action distributions (similar nodes and edges) would yield more honest evaluations of transfer learning in this domain, as there may be unintended bias introduced by Algorithm 1. Lastly, running an ablation study ranging across the values for $sparsity\_nodes$ and $sparsity\_edges$ would be valuable to pinpoint when the rich Reward Shaper overfits.

References

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[2] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*, 2017.

[3] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. *CoRR*, abs/1707.01476, 2017.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[5] Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. Prompt-learning for fine-grained entity typing. *CoRR*, abs/2108.10604, 2021.

[6] Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. Openprompt: An open-source framework for prompt-learning. *CoRR*, abs/2111.01998, 2021.

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

[8] Lihui Liu, Boxin Du, Yi Ren Fung, Heng Ji, Jiejun Xu, and Hanghang Tong. Kompare: A knowledge graph comparative reasoning system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3308–3318, 2021.

[9] Lihui Liu, Boxin Du, Jiejun Xu, Yinglong Xia, and Hanghang Tong. Joint knowledge graph completion and question answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1098–1108, 2022.

[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.

[11] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th ACL*, 2020.

[12] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

[13] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[14] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

[15] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32, 1992.

[16] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases, 2015.

[17] Liang Yao, Chengsheng Mao, and Yuan Luo. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193, 2019.