GHC: G: Deep Reinforcement Learning for Heterogeneous Relational Reasoning in Knowledge Graphs

Mandana Saebi (Advisor: Nitesh Chawla)
University of Notre Dame, USA
msaebi@nd.edu

ABSTRACT

Relational reasoning over knowledge graphs has become increasingly popular due to a variety of downstream applications such as question answering in dialogue systems, fact prediction, and recommendation systems. In recent years, reinforcement learning (RL) based solutions for knowledge graphs have been demonstrated to be more interpretable and explainable than other deep learning models. However, the current RL solutions still struggle with performance issues due to incomplete state representations and large action spaces for the RL agent. We address these problems by developing HRRL (Heterogeneous Relational reasoning with Reinforcement Learning), a type-enhanced RL agent that utilizes the local heterogeneous neighborhood information for efficient reasoning over knowledge graphs. HRRL improves the state representation using a graph neural network (GNN) for encoding the neighborhood information and utilizes entity type information for pruning the action space. Extensive experiments on real-world datasets show that HRRL outperforms state-of-the-art RL methods and discovers more novel paths during the training procedure, demonstrating the explorative power of our method.

1 INTRODUCTION

Relational reasoning is an important goal of machine learning and artificial intelligence [3, 4, 8, 17]. In the context of large-scale knowledge graphs (KG), relational reasoning addresses a number of important applications, such as question answering [2, 16], dialogue systems [9, 11], and recommender systems [1, 7, 20]. Most KGs are incomplete and the problem of inferring missing facts, or relational reasoning in KG, has become an increasingly important research topic. Reasoning in KGs can find new relational facts and predict missing facts, which is essential for accurate information retrieval and relation extraction from documents. Several previous works view this as a link prediction problem and attempt to solve it using network embedding and deep learning approaches [13, 18]. However, these methods are unable to discover multi-hop relations, and cannot provide a direct interpretation of their predictions. Lack of explainability in such models makes it hard to trust their prediction and ensure fairness and transparency when making critical business decisions. Recent advances in the area of deep reinforcement learning have inspired reinforcement learning (RL) based solutions for the KG reasoning problem [2, 6]. RL-based methods formulate the task of KG reasoning as a sequential decision-making process, in which the goal is to train an RL agent to walk over the graph by taking a sequence of actions (i.e., choosing the next entity) that connects the source to the target entity. The sequences of entities and relations can be directly used as a logical reasoning path for interpreting model predictions. For example, in order to answer the query *(Reggie Miller, plays sport, ?)*, the agent may find the following

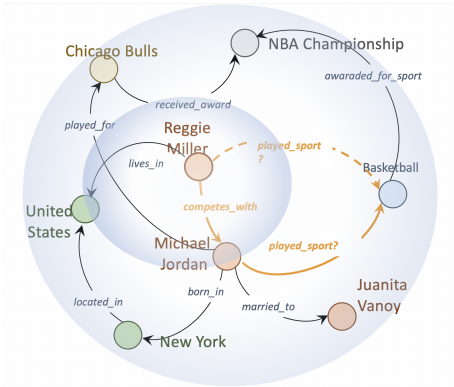

Figure 1: Given the query *(Reggie Miller, plays sport, ?)*, an RL-based solution may choose actions that lead to an incorrect answer, such as: *Reggie Miller* $\xrightarrow{\text{competes with}}$ *Michael Jordan* $\xrightarrow{\text{played for}}$ *Chicago Bulls*. HRRL is more likely to choose the correct path: *Reggie Miller* $\xrightarrow{\text{competes with}}$ *Michael Jordan* $\xrightarrow{\text{plays sport}}$ *Basketball* by paying attention to the entity type (entities with the same type are colored the same) and entity's neighborhood.

reasoning path in the KG: *Reggie Miller* $\xrightarrow{\text{competes with}}$ *Michael Jordan* $\xrightarrow{\text{plays sport}}$ *Basketball*. In this case *Reggie Miller*, *Michael Jordan*, and *Basketball* are all entities in the KG, and *competes with* and *plays sport* are relations. The agent is thus learning to navigate the entities and relations of the KG.

However, these RL solutions still face several challenges, including poor efficiency, due to large action space for the RL agent, and incomplete representation of the environment, in particular failing to capture the local neighborhood structure. We address these problems by introducing a Heterogeneous Relational reasoning with Reinforcement Learning (HRRL) agent that uses the heterogeneous neighborhood information for accurate relational reasoning over knowledge graphs. HRRL uses graph neural network (GNN) for encoding the neighborhood information and utilizes entity types to prune the action space. Our experiments on real-world datasets show that HRRL outperforms state-of-the-art RL methods.

2 APPROACH

In light of recent work on heterogeneous networks that have demonstrated the importance of heterogeneous information (represented as entity types in KGs) [5] and local neighborhood information [19] in graph mining, we take a broader approach. We propose to include entity type information in the state representation to help improving the search efficiency for the RL agent by taking more informed actions considering the heterogeneous context. We also

learn the heterogeneous neighborhood information simultaneously with training the RL agent to improve the predictions by enriching the state representation with local graph structure.

2.1 Data & Metrics

The experiments utilize three data sets presented in Table 1. Among the standard data sets used in the KG reasoning task, NELL-995 is the only one that explicitly encodes the entity types. Therefore, in addition to NELL-995, we incorporated two datasets Amazon e-commerce collection [15]. Each Amazon data contains a set of users, products, brands, and product categories. The full KG is represented by the number of *Facts* in Table 1. Before training, we partition *Facts* into a training set and a test set. We measure performance for each experiment with respect to Hits@k for k={1,5,10} and Mean Reciprocal Rank (MRR), which are standard KG reasoning metrics. Hits@k is measured as the percentage of test cases in which the correct entity e_d appears in the top k candidates in $\hat{E}_d$. MRR is a related metric, defined as the multiplicative inverse of the rank of the correct answer.

Dataset	$\lvert\mathcal{E}\rvert$	#Types	$\lvert\mathcal{R}\rvert$	#Facts
NELL-995	75,492	268	200	154,213
Amazon Beauty	16,345	5	7	52,516
Amazon Cellphones	13,837	5	7	31,034

Table 1: Description of the data sets used for our experiments. $\lvert\mathcal{E}\rvert$ describes the total number of nodes in the KG, Types describes the number of entity types, $\lvert\mathcal{R}\rvert$ describes the number of edge types, and Facts describes the total number of edges.

2.2 Model

Suppose we have a knowledge graph $\mathcal{G} = \{(e_s, r, e_d)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ where $\mathcal{E}$ is a set of entities and $\mathcal{R}$ is a set of relations. Given a query $(e_s, r, ?)$, e_s is called the source entity and r is the query relation. Our goal is to predict the target entity $e_d \in \mathcal{E}$. In KGs, we are not only interested in accurate prediction of the target entity e_d, but also understanding the reasoning path the model uses to predict e_d. Rather than treating the task as a form of link prediction, RL models instead train an agent to traverse the nodes of a KG via logical reasoning paths. Below we provide the details of RL formulations of the problem. Similar to [2, 6, 16], we formulate this problem as a Markov Decision Process (MDP), in which the goal is to train a policy gradient agent (using REINFORCE [14]) to learn an optimal reasoning path to answer a given query. We express the RL framework as a set of states, actions, rewards, and transitions. An overview of the model is displayed in Figure 2.

States. The state s_t at time t consists of the input query, the entity representation at which the agent is located at time t and the history of the entities and relations traversed by the agent until time t. The history is encoded using a 2-layer Long-Short-Term-Memory network (LSTM). In this solution, we propose to enrich the state representation with entity type and neighborhood information, which is explained later below.

Actions. At each time-step, the action is to select an edge (and move to the connecting entity) or stay at the current entity. Therefore, the action space consists of all immediate neighbors of the current node e_t, and the node itself.

Rewards. The agent evaluates each action and chooses the one that will maximize a reward. Inspired by [6], we use pre-trained KG embeddings based on existing KG embedding methods to design a soft reward function for the terminal state.

Below we explain our modeling of entity type embeddings for action space pruning and a heterogeneous neighbor encoder:

Action space pruning using entity type. Real-world KGs contain huge numbers of entities and relations. As a result, the RL agent often encounters nodes with large out-degrees. In these cases, exploring the possible paths to determine the optimal action is computationally expensive. We argue that the type information can be helpful for reducing the action space, especially for nodes with a high out-degree. Entity type information can be used to limit the search only to the entities that are best matching the previously visited entities and actions. To achieve this, we measure the similarity of all possible actions given the entity type embedding of current and possible target entities and only keep the top n candidates. In order to build the entity type representation, we propose to aggregate the vector representation of the entities with a similar type using a mean / max pooling layer. We concatenate the type embeddings with entity embeddings to build the type-enhanced entity embeddings.

Heterogeneous neighbor encoder. After generating the type embeddings, we feed the type-enhanced embeddings together with the relation embeddings to the heterogeneous neighbor encoder to generate the enriched entity representation. Recent studies [19] have demonstrated that explicitly encoding graph local structure can benefit entity embedding learning in knowledge graphs. Inspired by this motivation, we propose a Graph Neural Network (GNN) to learn the enriched entity embedding by aggregating entity's neighborhood information. GNNs are neural network models that directly operate on the graph and have shown impressive success in capturing the node neighborhood with arbitrary depth [19]. This heterogeneous neighbor-encoder learns the optimal neighborhood structure for the relational reasoning task while considering the heterogeneous content of the KG. The output of this module will be the entity representation in the RL model.

2.3 Experimental setup

We compare HRRL against state-of-the-art RL baselines: MINERVA (agent-based) [2], and Lin et al [6]. We also tried different variations of HRRL by removing either of the model components. The type-enhanced embeddings are removed in HRRL (-T) the heterogeneous neighbor encoder is removed in HRRL (-N). For NELL-995, we utilize the same hyperparameters described in [6] and [2] when training Lin et al. and MINERVA, respectively. For the two Amazon datasets, we perform a grid search for HRRL and all the baselines and report the best performance for each. For all datasets, we train the KG embedding models (DistMult [18] and ComplEx [13]) for 1000 epochs each to obtain the pre-trained inputs for the RL agent. We initially tested with different embedding and pooling methods for the pre-trained embeddings and settled on those that achieved the best performance: ComplEx for NELL-995 and Distmult for both Amazon data sets. We generated the type embeddings using

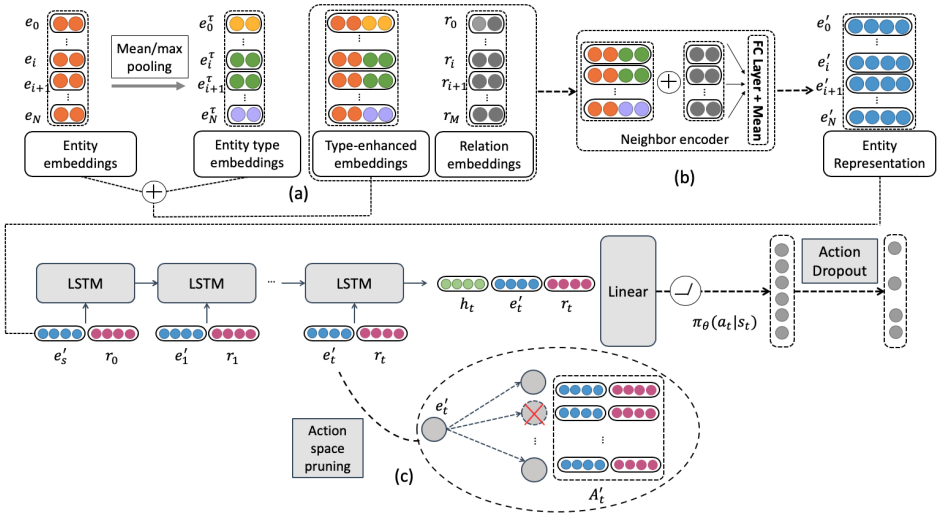

Figure 2: Model overview. (a) The type embeddings are first created by max/mean pooling on the entities with a similar type. The type embeddings are then concatenated with the entity embeddings to create the type-enhanced embeddings. (b) The type-enhanced embeddings are then passed to the neighbor encoder to create the final entity representation fed to RL. (c) The action-space is pruned using the type-enhanced entity embeddings.

Data Set	NELL-995				Amazon Beauty				Amazon Cellphones			
Metric (%)	@1	@5	@10	MRR	@1	@5	@10	MRR	@1	@5	@10	MRR
Lin et. al [6]	65.6	80.4	84.4	72.7	20.6	33.6	39.5	27.1	12.2	22.2	27.6	17.5
MINERVA [2]	59.8	79.5	82.1	68.9	17.5	29.3	38.2	24.3	6.8	12.3	22.7	11.6
HRRL (-T)	66.9	81.9	85.2	74.1	21.2	34.6	40.5	27.9	12.6	22.9	28.1	17.9
HRRL (-N)	67.1	81.8	85.1	73.1	20.7	33.8	39.6	27.2	12.3	22.4	27.9	17.6
HRRL	**68.9**	**83.2**	**86.7**	**74.8**	**21.8**	**35.1**	**40.7**	**28.2**	**12.9**	**23.1**	**28.5**	**18.2**

Table 2: Experimental results on the NELL-995, Amazon Beauty, and Amazon Cellphone data sets. @{1, 3, 5, 10} and MRR are standard KG reasoning metrics and are described in Section 2.1. We compare our method to state-of-the-art RL-based methods (MINERVA, Lin et al.). Bolded numbers indicate the best-performing method.

max-pooling for the NELL-995 dataset and mean-pooling for both Amazon datasets.

3 RESULTS AND CONTRIBUTIONS

3.1 KG reasoning

Our experimental results are described in Table 2. For NELL-995 data, we quote the baseline results reported in [2, 6]. In all three datasets HRRL outperforms both RL baselines (Lin et al. [6] and MINERVA [2]). However, in the Amazon datasets, the performance of all methods is significantly lower. HRRL results in a 4% improvement in MRR (and 5.43% in Hits@1) over the best RL baseline on Amazon Cellphones and a 3.9% improvement in MRR (and 5.5% in Hits@1) on Amazon Beauty. On the NELL-995 dataset, HRRL results in 2.8% improvement in MRR and 4% improvement in Hits@1 over the best performing baseline. We also performed ablations studies to analyze the effect of each module in our model. We notice that removing the heterogeneous neighbor encoder results in a higher drop in performance in the Amazon datasets. This gap is quite smaller in the NELL-955 data. Our results show that pruning the action space based on the entity type information (compared to pruning based on entity page rank, as done in [6]) results in a better performance on the Amazon datasets. We believe due to the sparsity of these two knowledge graphs, type information was more effective for action space pruning than entity page ranks, as done in [6]. Note that, there are only 5 entity types in the Amazon datasets. As a result, the number of entities that will be discarded (due to type mismatch) is higher which assists the agent to discover a better path.

3.2 Case studies

In this section, we present a few case studies that show the strength of our proposed method. We first focus on the NELL-995 dataset which has a much larger set of entity types. Table 3 shows top frequent path types discovered by HRRL and the best performing RL baseline ([6]) for a few example queries in the NELL-995 data. In this table p_1 and p_2 show the occurrence probability for each path type discovered by HRRL and [6], respectively. We notice that our method is more successful in discovering better paths as it takes advantage of the heterogeneous content. As an example, for the query (Knicks, *team plays sport*, ?), our method discovers the path:

Query	Reasoning path types	p_1(%)	p_2(%)
e_s: Knicks	sports team $\xrightarrow{\textit{team plays against team}}$ sportsteam $\xrightarrow{\textit{athlete plays for team}^{-1}}$ athlete $\xrightarrow{\textit{plays sport}}$ sport	5.92	3.19
r: *plays sport*	sports team $\xrightarrow{\textit{team home stadium}}$ event venue $\xrightarrow{\textit{sport uses stadium}^{-1}}$ sport $\xrightarrow{\textit{plays sport}^{-1}}$ sports team	5.77	5.19
e_t: ?	sports team $\xrightarrow{\textit{sports game team}^{-1}}$ sports game $\xrightarrow{\textit{sports game team}}$ sports team	3.79	2.28
e_s: Amazon	company $\xrightarrow{\textit{acquired}}$ company $\xrightarrow{\textit{company sector}}$ economic sector	4.88	3.52
r: *company*	company $\xrightarrow{\textit{has office in city}}$ city $\xrightarrow{\textit{has office in city}^{-1}}$ company $\xrightarrow{\textit{has office in city}}$ city	3.71	4.01
sector	company $\xrightarrow{\textit{has office in city}}$ city $\xrightarrow{\textit{has office in city}^{-1}}$ newspaper $\xrightarrow{\textit{journalist works for}^{-1}}$ journalist	2.93	2.32
e_t: ?			
e_s: MSU	university $\xrightarrow{\textit{person belongs to org}^{-1}}$ coach $\xrightarrow{\textit{works for}}$ sports team $\xrightarrow{\textit{located in state}}$ state	22.65	7.41
r: *located in*	university $\xrightarrow{\textit{person belongs to org}^{-1}}$ coach $\xrightarrow{\textit{works for}}$ sports team $\xrightarrow{\textit{team plays against team}}$ sports team	10.21	8.64
state	university $\xrightarrow{\textit{is acronym for}}$ university $\xrightarrow{\textit{language of university}^{-1}}$ language $\xrightarrow{\textit{language of country}}$ country	7.52	28.4
e_t: ?			

Table 3: Top 3 frequent paths for our method ($P1$) and Lin et al. ($P2$) along with the occurrence probability. Note that for each query, the number of all possible path types can vary between 73-277 in these examples. As a result, the probability of each path type is low.

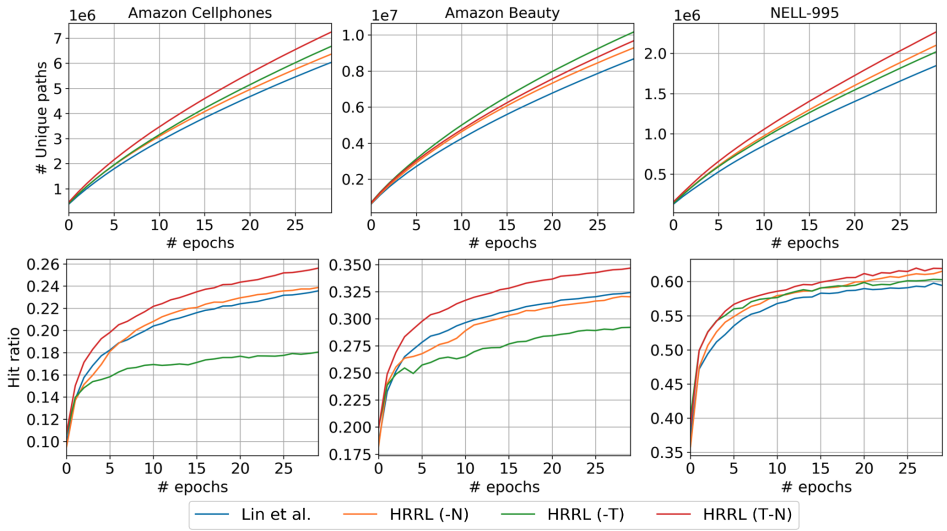

Figure 3: Performance of our model on the development set at different training epochs. The top figures show the number of unique paths visited during each epoch. The bottom rows show the hit ratio for the entire development set.

Knicks (sports team) [67] $\xrightarrow{\textit{team plays against team}}$ Trail Blazers (sports team) [30] $\xrightarrow{\textit{athlete plays for team}^{-1}}$ Steve Blake (athlete) [4] $\xrightarrow{\textit{plays sport}}$ basketball (sport) [242], in which the underlined number in the bracket shows the entity's out-degree. This path matches the first row in Table 3 for this query, which has the highest probability of selection by HRRL. Lin et al. [6] method, on the other hand, is more likely to select the second path type, which cannot reach the correct answer.

As another example, we consider the query: (Amazon, *company sector*, ?) for which our method discover the path: Amazon (company) [4] $\xrightarrow{\textit{acquired}}$ AbeBooks (company) [2] $\xrightarrow{\textit{company sector}}$ internet (economic sector) [26]. Therefore, HRRL is able to infer the company sector using the sector of the company that was acquired by Amazon. Table 3 shows the other possible path types. We can see that Lin et al. [6] method has higher probability of selecting

an incorrect path which leads to a (city) entity, as opposed to an (economic sector) entity.

The third query in Table 3 shows the top path types for query: (MSU, *located in state*, ?), HRRL discovers the following path: MSU (university) [5] $\xrightarrow{\textit{person belongs to org}^{-1}}$ Tom Izzo (coach) [2] $\xrightarrow{\textit{works for}}$ Michigan State (sports team) [23] $\xrightarrow{\textit{located in state}}$ Michigan (state) [137], corresponding to the first row for the fourth query in Table 3. Lin et al. [6] method on the other hand, is more likely to select a path type that leads to a (country) or a (sport team) entity, as opposed to a (state) entity. For this query, number of all possible path types is smaller (75 for HRRL and 73 for [6]), as a result, the top frequent paths have relatively larger probabilities.

3.3 Path diversity and convergence

In order to show the explorative power of our RL agent, we compare the number of unique paths discovered from the development set during the training procedure. Figure 3 shows that path diversity (top row) improves across all models as the model performance (bottom row) improves. For this analysis, we compare our ablation models (HRRL (-N) and HRRL (-T)) with the best performing RL baseline by Lin et al. [6]. Our method is more successful in discovering novel paths and obtains a better hit ratio on the development set. On the Amazon Beauty data, the number of unique paths discovered by HRRL (-T) is higher than both combined (Ours) while in Amazon Cellphones the combined model performs better, but similar to Amazon Beauty, HRRL (-N) performs better than HRRL (-T). NELL-955 shows a different trend where removing the type information results in a larger drop in the number of unique paths, compared to the heterogeneous neighbor encoder. This is intuitive, since NELL-955 contains far more entity type than Amazon datasets, and inclusion of type information may be a positive factor for discovering new paths. In terms of convergence, Amazon Beauty and Amazon Cellphones show a similar trend, and removing the type information significantly reduces the hit ratio. This gap is smaller for NELL-995 data, though our model still shows improvement in hit ratio on this dataset.

4 RELATED WORK

Relational reasoning over knowledge graphs has attracted significant attention over the past few years. Recent works [5, 18] approached this problem by embedding the relation and entities into a vector space and identifying related entities by similarity in the vector space. However, these methods have some important drawbacks, including:(1) They cannot perform multi-hop reasoning. That is, they only consider pairwise relationships and cannot reason along a path. (2) They cannot explain the reasoning behind their predictions. Because they treat the task as a link prediction problem, the output of their prediction is a probabilistic value, indicating whether a relation may exist or not.

With the recent success of deep reinforcement learning in AlphaGO [12], researchers began to adopt RL to solve a variety of problems that were conventionally addressed by deep learning methods. As a result, more recent methods proposed using RL to solve the relational reasoning problem in knowledge graphs [2, 6, 16]. Deeppath [16] was the first method that used RL to find relation paths between two entities in KGs. MINERVA [2] learns to do relational reasoning by jointly selecting an entity-relation pair via a policy network. Lin et al. [6] implement reward shaping to address the problem of the sparse reward signal and action dropout to reduce the effect of incorrect paths. Because these RL models treat the KG reasoning problem as a path reasoning problem, they are able to overcome both drawbacks of embedding methods that are outlined above. However, the RL models have drawbacks of their own, the most notable of which are computational cost and predictive accuracy.

5 CONCLUSION

We introduced HRRL for improving the performance of path-based reasoning using reinforcement learning. HRRL addresses the key challenges stemming from large action spaces and accurate representation of an entity's heterogeneous neighborhood. The key contributions of HRRL include: (1) an expressive vector representation for heterogeneous entity type-embeddings, (2) pruning the action space for improving the choice of next actions, and (3) leveraging a GNN for incorporating the neighborhood information.

We evaluate HRRL on three contemporary datasets. Our results show that incorporating information about the heterogeneous neighborhood results in improved performance for the query answering task. We show that the type of information is important for faster convergence and finding more diverse paths, and the neighborhood information improves the performance on unseen queries.

6 ACKNOWLEDGEMENTS

This work is part of the research lead by the author available at [10]. The author has presented her own contribution in this proposal. All the collaborators in [10] are aware of this submission and have read and approved this proposal.

REFERENCES

[1] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 456–464. ACM, 2019.

[2] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola, and A. 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] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

[4] D. Koller, N. Friedman, S. Džeroski, C. Sutton, A. McCallum, A. Pfeffer, P. Abbeel, M.-F. Wong, D. Heckerman, C. Meek, et al. *Introduction to statistical relational learning*. MIT press, 2007.

[5] K. Lei, J. Zhang, Y. Xie, D. Wen, D. Chen, M. Yang, and Y. Shen. Path-based reasoning with constrained type attention for knowledge graph completion. *Neural Computing and Applications*, pages 1–10, 2019.

[6] X. V. Lin, R. Socher, and C. Xiong. Multi-hop knowledge graph reasoning with reward shaping. *arXiv preprint arXiv:1808.10568*, 2018.

[7] F. Liu, R. Tang, X. Li, W. Zhang, Y. Ye, H. Chen, H. Guo, and Y. Zhang. Deep reinforcement learning based recommendation with explicit user-item interactions modeling. *arXiv preprint arXiv:1810.12027*, 2018.

[8] S. Muggleton. Inductive logic programming. *New generation computing*, 8(4):295–318, 1991.

[9] B. Peng, X. Li, L. Li, J. Gao, A. Celikyilmaz, S. Lee, and K.-F. Wong. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. *arXiv preprint arXiv:1704.03084*, 2017.

[10] M. Saebi, S. Krieg, C. Zhang, M. Jiang, and N. Chawla. Heterogeneous relational reasoning in knowledge graphs with reinforcement learning. *arXiv preprint arXiv:2003.06050*, 2020.

[11] I. V. Serban, C. Sankar, M. Germain, S. Zhang, Z. Lin, S. Subramanian, T. Kim, M. Pieper, S. Chandar, N. R. Ke, et al. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*, 2017.

[12] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[13] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.

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

[15] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, and Y. Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. *arXiv preprint arXiv:1906.05237*, 2019.

[16] W. Xiong, T. Hoang, and W. Y. Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv preprint arXiv:1707.06690*, 2017.

[17] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Infinite hidden relational models. *arXiv preprint arXiv:1206.6864*, 2012.

[18] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[19] C. Zhang, L. Yu, M. Saebi, M. Jiang, and N. Chawla. Few-shot multi-hop relation reasoning over knowledge bases. In *Proceedings of Conference on Empirical Methods in Natural Language Processing: Findings*, pages 580–585, 2020.

[20] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, pages 167–176. International World Wide Web Conferences Steering Committee, 2018.