# Hierarchical Multihop Reasoning on Knowledge Graphs 

Zikang Wang ©, Institute of Automation, Chinese Academy of Sciences, Bejing, 100864, China, and also with the University of Chinese Academy of Sciences, Beijing, 100049, China
Linjing Li © and Daniel Dajun Zeng, Institute of Automation, Chinese Academy of Sciences, Bejing, 100864, China, also with the University of Chinese Academy of Sciences, Beifing, 100049, China, and also with the 4Data Science Institute (Longhua), Shenzhen, 518129, China


#### Abstract

Multihop knowledge reasoning aims to find missing entities for incomplete triples by finding paths on knowledge graphs. It is a fundamental and important task. In this article, we devise a hierarchical reinforcement learning algorithm to model the reasoning process more effectively. Unlike existing methods directly reason on entities and relations, we adopt a high-level reasoning layer to deal with abstract concepts, which guides the reasoning process conducted at the low level for concrete entities and relations. Our approach yields competitive results on link prediction on both NELL-995 and FB15k-237 datasets. The comparison to baselines also demonstrates the effectiveness of the hierarchical structure.


Knowledge graphs store knowledge in the form of graphs and provide a solid foundation for various applications of artificial intelligence. However, knowledge graphs suffer from the problem of incompleteness, which harms the performance of downstream applications. Knowledge graph reasoning is an important task dedicated to addressing this problem.

The goal of the knowledge graph reasoning is completing a given triple ( $h, r$, ?), i.e., find the tail entity $t$ that satisfies the relation $r$ with head entity $h$. In this article, we situate our study in multihop reasoning, which aims to discover missing links based on several known links in the knowledge graph. For example, for head entity "Kurt Cobain" and relation "person language" to learn the tail entity "English," the multihop reasoning models find three links in knowledge graph: (Kurt Cobain, born in, Washington), (Washington, located in, United States), (United States, country speak, English). Thus, the multihop reasoning model learns the missing link (Kurt Cobain, person language, English).

Existing methods start directly with the head entity and find the target entity by finding paths on the knowledge graph. However, this approach cannot model hierarchical reasoning. For example, in Figure 1, a natural way for humans to find paths with semantic "person language" is to decompose the target "person language" into a combination of several concepts that may not exist in the knowledge graph, like "nationality" and "language," then find actual reasoning paths that satisfy these concepts as much as possible according to the specific entities and edges in the knowledge graph.

The goal of this article is to simulate the hierarchical reasoning process by using a two-level structure. We conduct reasoning on concrete entities at the low level, which is guided by high-level reasoning on embeddings of abstract concepts. In contrast to existing methods that deal directly with entities and relations, we find paths by first computing an abstract relation embedding as the goal, transitions between concrete entities are based on this embedding. Hierarchical reinforcement learning (HRL) can model this process perfectly. ${ }^{1}$ Therefore, we propose an HRLbased reasoning model to address this issue.

Specifically, for a query ( $h, r$, ?), starting with $h$, we first learn a high-level strategy to reason on the abstract embeddings and determine whether the target entity is reached. If not, we compute the

[^0]single layer reasoning: (Kurt Cobain, personLanguage, ?)

hierarchical reasoning: (Kurt Cobain, personLanguage, ?)


FIGURE 1. Example of hierarchical reasoning.
embedding of the next chosen relationship and proceed to the low-level reasoning. We set this relation embedding as the goal of the low-level reasoning, where we select the concrete paths in the knowledge graph. Once the goal is reached, then we return to the high level to continue the previous process until the target entity $t$ is reached.

We conducted the link prediction experiment on two large datasets and obtained outstanding results. The comparison with the single-level reinforcement learning ( RL ) approach also demonstrates the effectiveness of the proposed hierarchical structure.

## RELATED WORK

Multihop reasoning is proposed to address the drawback of embedding-based reasoning on knowledge graphs. Embedding-based reasoning methods treat triples separately, and reasoning based on the learned embeddings of entities and relations. This kind of methods include TransE, ${ }^{2}$ TransR, ${ }^{3}$ DistMult, ${ }^{4}$ Com$\mathrm{plEx},{ }^{5}$ etc.

Different from embedding-based reasoning, multihop reasoning is based on paths instead of separate triples. One class of methods treats the paths in knowledge graphs as sequences and infer relations based on the semantic embeddings learned from the paths. These methods include PRA, ${ }^{6}$ Compositional Reasoning, ${ }^{7}$ RNN-Chains, ${ }^{8}$ etc. PRA ${ }^{6}$ is the first multihop reasoning model, which reasons in discrete space and hard to generalize. Compositional reasoning and

RNN-Chains extend PRA to continuous space, and improves the generalization ability.

The other class of methods is based on RL. They frame the paths as Markov Decision Process and solve the problem using RL, such as DeepPath, ${ }^{9}$ MINERVA, ${ }^{10}$ and AttnPath. ${ }^{11}$ Compared to models in the first class, RL-based models can control the properties of the found paths and learn long chains of reasoning. However, existing methods only reason on specific entities and relations. Using MINERVA as an example, it starts from the head entity, transferred to the next entity by selecting the optimal relation at each step. It cannot model hierarchical reasoning and its reasoning ability is limited.

Hierarchical RL, ${ }^{1}$ as an improved model of RL, can model multilevel decision-making processes in two levels, which is a good simulation of hierarchical reasoning. In this article, we attempt to reason on knowledge graphs using hierarchical RL.

## METHODOLOGY

In this section, we introduce the HRL-based reasoning model. The overall structure of our model is shown in Figure 2.

## Problem Definition

A knowledge graph $\mathcal{G}$ is a collection of triples, $\mathcal{G}=$ $\{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ are sets of entities and relations. A triple ( $h, r, t$ ) indicates that head entity $h$ and tail entity $t$ have relation $r$. This article


FIGURE 2. Overall architecture of our model, using the same example as in Figure 1.
aims to settle the problem ( $h_{q}, r_{q}$, ?), which is finding the missing tail entity $t_{q}$ among $\mathcal{E}$. It is more difficult than fact prediction task, which aims to determine whether a triple $\left(h_{q}, r_{q}, t_{q}\right)$ is true. We represent the reasoning process as a Markov decision process and address it using the REINFORCE algorithm.

## Hierarchical RL Formulation

Given a query ( $h_{q}, r_{q}$, ?), we conduct reasoning at two levels in each step. At the high level, we determine whether the target entity is reached, end the reasoning if so. Otherwise, we compute the embedding of the relation to be taken in the next step and proceed to the low-level reasoning. At the low level, we reason on the concrete entities with the embeddings learned from high level as the goal. To facilitate the above iteration, we add an inverse relation $r^{-1}$ for each $r$ to make it possible to return to previous states. We also add a self-loop relation for each entity, which leads to the stop of reasoning.

We describe this process as a hierarchical semi-Markov decision process: 1) the high-level RL process models the transitions between abstract embeddings; and 2) the low-level RL process models the transitions between concrete entities in the given knowledge graph.

## High-Level RL

High-level reasoning simulates state transits between abstract concept states. At the high level, reasoning starts from the head entity $h_{q}$ with goal $r_{q}$.

States: The state of the agent at time step $t$ should include the head entity $h_{q}$ and relation $r_{q}$ in the query to guide the reasoning process, as well as the answer entity $t_{q}$ to assign rewards and the current entity $e_{t}$ at time step $t$. Thus, the overall state at time step $t$ is defined as $s_{t}^{h}=\left(h_{q}, r_{q}, t_{q}, e_{t}\right)$.

Observations: The environment is partially observable as the answer entity is not known during the reasoning. Then, the observation function $\mathcal{O}^{h}$ at high level is defined as $o_{t}^{h}=\mathcal{O}^{h}\left(s_{t}^{h}=\left(h_{q}, r_{q}, t_{q}, e_{t}\right)\right)=\left(h_{q}, r_{q}, e_{t}\right)$, where $o_{t}^{h}$ is the observation at time step $t$ at high level.

Actions: At the high level, the agent is responsible for determining whether or not the answer entity has been reached. The set of possible actions is $\mathcal{A}^{h}=$ $\{0,1\}$, where $\mathcal{A}^{h}$ is the action space in high level. Denote $a_{t}^{h}$ as the action at time step $t$ in high level, $a_{t}^{h} \in \mathcal{A}^{h}$. If the target entity has been reached, then ends the reasoning with $a_{t}^{h}=0$, and the current entity is the found target entity. Otherwise $a_{t}^{h}=1$, the agent enters the low-level reasoning.

Transition: If $a_{t}^{h}=0$, the agent finds the target entity and the reasoning process ends; otherwise, the agent enters the low-level reasoning, which focuses on the transitions on concrete entities and starts at entity $e_{t}$ with goal $r_{t}^{h}$. $r_{t}^{h}$ is the action embedding and will be explained later. In this case, the transition function is defined as $\sigma^{h}\left(s_{t}^{h}\right)=s_{t}^{l}$, where $\sigma^{h}(\cdot)$ is the transition function in the high level, $s_{t}^{h}$ and $s_{t}^{l}$ denotes the state at $t$ in high level and low level, respectively.

Rewards: The reward $R^{h}$ is set based on whether the target entity the agent chosen is the correct
answer entity $t_{q}$. If the final entity $e_{T}$ is correct, i.e., $e_{T}=t_{q}$, the reward is set to be +1 , otherwise the reward is 0

$$
R^{h}=\left\{\begin{array}{lc}
1, & \text { if } e_{T}=t_{q}  \tag{1}\\
0, & \text { otherwise }
\end{array}\right.
$$

## Low-Level RL

The low-level reasoning is concerned with the transition of states between specific entities and relations. At the low level, reasoning starts from the current entity $e_{t}$ with goal $r_{t}^{h}$, where $e_{t}$ indicates that the agent is at entity $e$ at time step $t$, and $r_{t}^{h}$ is the action embedding from high level and set as the query embedding in low level, which will be discussed later.

States: The agent reasons at the low level by varying between specific entities. The low-level reasoning is considered as an independent RL process whose time step is denoted as $t^{\prime}$. The state of the agent at time step $t^{\prime}$ should include the query embedding $r_{t}^{h}$ to guide the reasoning process, as well as the specific entity $e_{t^{\prime}}$ at time step $t^{\prime}$. Thus, after entering the low level at time $t$, the overall state at time step $t^{\prime}$ in low level is defined as $s_{t^{\prime}}^{l}=\left(e_{t^{\prime}}, r_{t}^{h}\right)$.

Observations: At low level, observations are the same as states, i.e., $o_{t^{\prime}}^{l}=\mathcal{O}^{l}\left(s_{t^{\prime}}^{l}=\left(e_{t^{\prime}}, \boldsymbol{r}_{t}^{h}\right)\right)=\left(e_{t^{\prime}}, \boldsymbol{r}_{t}^{h}\right)$, where $o_{t^{\prime}}^{l}$ is the observation at time $t^{\prime}$ in low level, and $\mathcal{O}^{l}$ is the observation function at low level.

Actions: At the low level, the agent transfers between specific entities to find the target entity. The action space $\mathcal{A}^{l}$ includes all the outgoing edges of the entity. $\mathcal{A}^{l}$ at time step $t^{\prime}$ consists of all the relations and entities at time $t^{\prime}+1$, then $\mathcal{A}_{t^{\prime}}^{l}=$ $\left\{\left(r_{t^{\prime}+1}, e_{t^{\prime}+1}\right) \mid\left(e_{t^{\prime}}, r_{t^{\prime}+1}, e_{t^{\prime}+1}\right) \in \mathcal{G}\right\}$. Note that we add a self-loop edge for each entity. Choosing the self-loop edge indicates the end of the reasoning, and the chosen target entity is the current entity.

Transition: In low-level RL, states transit between entities by selecting relations outgoing from current entity $e_{t^{\prime}}$. The query embedding $r_{t}^{h}$ stays unchanged during the transition. The transition function is defined as $\sigma^{l}\left(e_{t^{\prime}}, \boldsymbol{r}_{t}^{h}\right)=\left(e_{t^{\prime}+1}, \boldsymbol{r}_{t}^{h}\right)$, where $\sigma^{l}(\cdot)$ is the transition function in low level.

Rewards: At low level, no direct feedback can be obtained. In order to evaluate its reasoning performance, we use the score function in the representation learning method. We assume the low-level reasoning starts from entity $e_{t_{0}^{\prime}}$ and reaches tail entity $e_{t_{T}^{\prime}}$, the representation of the path corresponds to the query embedding $r_{t}^{h}$. Then, the score function is used to calculate reward $R^{l}$. Take model TransE as an example, the low-level reward $R^{l}$ is set as

$$
\begin{equation*}
R^{l}=-\left\|e_{t_{0}^{\prime}}+r_{t}^{h}-e_{t_{T}^{\prime}}\right\|^{2} \tag{2}
\end{equation*}
$$

We use this setting in our model, it can be flexibly replaced by other more complicated score functions.

## Policy Network

At step $t$ at the high level, we use a history embedding $h_{t}^{h}$ to store the historical information to help choose action. The history embedding $h_{t}^{h}$ is computed based on the previous history embedding $h_{t-1}^{h}$, previous relation embedding $r_{t-1}^{h}$, and the embedding of current observation $o_{t}^{h}$

$$
\begin{equation*}
\boldsymbol{h}_{t}^{h}=\mathbf{L S T M}\left(\boldsymbol{h}_{t-1}^{h},\left[\boldsymbol{r}_{t-1}^{h} ; \boldsymbol{o}_{t}^{h}\right]\right) \tag{3}
\end{equation*}
$$

where LSTM is the long short-term memory network.
Then, we apply a policy network to compute the embedding $r_{t}^{h}$ of the next step. At high level, $r_{t}^{h}$ is fed into an MLP classifier to determine whether to end the reasoning

$$
\begin{gather*}
\boldsymbol{r}_{t}^{h}=\boldsymbol{\operatorname { R e L }} \mathbf{U}\left(W\left[\boldsymbol{h}_{t}^{h} ; \boldsymbol{o}_{t}^{h}\right]\right)  \tag{4}\\
a_{t}^{h}=\operatorname{MLP}\left(r_{t}^{h}\right) \tag{5}
\end{gather*}
$$

where $W$ is a parameter matrix.
At the low level, a relation is chosen according to a similar process, the history embedding $h_{t^{\prime}}^{l}$ at time $t^{\prime}$ in low level is calculated as

$$
\begin{equation*}
\boldsymbol{h}_{t^{\prime}}^{l}=\mathbf{L S T M}\left(h_{t^{\prime}-1}^{l},\left[\boldsymbol{r}_{t^{\prime}-1} ; \boldsymbol{o}_{t^{\prime}}^{l}\right]\right) \tag{6}
\end{equation*}
$$

where $r_{t^{\prime}-1}$ is the embedding of the specific relation chosen at time step $t^{\prime}-1$.

The specific relation is chosen according to
$a_{t^{\prime}}^{l} \sim \operatorname{Categorical}\left(\operatorname{softmax}\left(A_{t} \operatorname{ReLU}\left(W\left[\boldsymbol{h}_{t^{\prime}}^{l} ; \boldsymbol{o}_{t^{\prime}}^{l}\right]\right)\right)\right)$
where $a_{t^{\prime}}^{l}$ is corresponding to the chosen relation and $A_{t}$ is the stacked embeddings of relations in $\mathcal{A}_{t^{\prime}}^{l}$.

## Training

In the training phase, we aim to find optimal parameters $\theta^{h}, \theta^{l}$ to maximize both rewards in low level and high level

$$
\begin{align*}
J^{h}\left(\theta^{h}\right) & =\mathbb{E}_{a_{1}, a_{2}, \ldots, a_{T-1} \sim \pi^{h}}\left[R^{h}\left(e_{T}\right)\right]  \tag{8}\\
J^{l}\left(\theta^{l}\right) & =\mathbb{E}_{a_{1}^{\prime}, a_{2}^{\prime}, \ldots, a_{T-1}^{\prime} \sim \pi^{l}}\left[R^{l}\left(e_{t_{T}^{\prime}}\right)\right] \tag{9}
\end{align*}
$$

where $\theta^{h}, \theta^{l}$ are parameters of high level and low level, respectively, $\pi^{h}, \pi^{l}$ are the policies of high level and low level, respectively.

REINFORCE ${ }^{12}$ algorithm is employed to optimize the rewards. The training algorithm is shown in Algorithm 1.

```
Algorithm 1. Training Algorithm
    Initialize \(\theta^{h}, \theta^{l}\)
    for episode \(\leftarrow 1\) to \(N\) do
        Create a path \(p^{h}\)
        Initialize history embedding \(\quad \boldsymbol{h}_{0} \leftarrow 0\)
        for \(i \leftarrow 1\) to \(\left|p^{h}\right|\) do
            Calculate history embedding \(\boldsymbol{h}_{t}^{h} \leftarrow \mathbf{L S T M}\left(h_{t-1}^{h}\right.\),
            \(\left.\left[r_{t-1}^{h} ; \boldsymbol{o}_{t}^{h}\right]\right)\)
        Create a path \(p^{l}\)
        for \(j \leftarrow 1\) to \(\left|p^{l}\right|\) do
            Calculate reward \(R^{l}\left(p_{j}^{l}\right)\)
                Calculate gradient \(\nabla J^{l} \leftarrow-\frac{\partial}{\partial \theta^{l}} \log \pi^{l}\)
                \(\left(e_{i}^{j} \mid e_{i-1}, r_{i-1}^{h}\right)\)
            end for
        Calculate reward \(R^{h}\left(p_{i}^{h}\right)\)
        Calculate gradient \(\nabla J^{h} \leftarrow-\frac{\partial}{\partial \theta^{h}} \log \pi^{h}\left(a_{i} \mid h_{i}^{h}, \boldsymbol{o}_{i}^{h}\right.\),
        \(a_{i-1}, \theta^{l}\) )
        end for
        Update \(\theta^{h}, \theta^{l}\) using \(\nabla J^{h}, \nabla J^{l}\)
        end for
```


## EXPERIMENTS

In this section, we evaluate the model on two large datasets. The experimental results show the effectiveness of the hierarchical structure.

## Dataset

We employ NELL-995 ${ }^{13}$ and FB15k-237, ${ }^{14}$ to evaluate both of them are widely adopted datasets. NELL-995 is extracted from a never-ending language learning (NELL) system. FB15k-237 is from Freebase, which provides general facts of the world. NELL-995 has 12 subtasks while FB15k-237 has 20. Each of these subtasks is composed of triples containing the same relation. Statistics in detail are shown in Table 1.

## Baselines and Implementation Details

Ten methods are included in the set of baselines. Of which, five are embedding-based models including TransE, ${ }^{2}$ TransR, ${ }^{3}$ DistMult, ${ }^{4}$ ComplEx, ${ }^{5}$ and ConvE. ${ }^{15}$ The other five are multihop reasoning methods, including PRA, ${ }^{6}$ RNN-Chain, ${ }^{8}$ DeepPath, ${ }^{9}$ MINERVA, ${ }^{10}$ and AttnPath. ${ }^{11}$ Our reinforcementbased learning approach belongs to multihop reasoning, and thus multihop reasoning methods are the focus of comparison. Among them, MINERVA is

TABLE 1. Statistics of datasets FB15k-237 and NELL-995.

| Dataset | $\|\mathcal{E}\|$ | $\|\mathcal{R}\|$ | \#Triples | \#Tasks |
| :--- | :---: | :---: | :---: | :---: |
| FB15k-237 | 14,505 | 237 | 310,116 | 20 |
| NELL-995 | 75,492 | 200 | 154,213 | 12 |

a single-level reasoning model based on RL, the comparison with MINERVA shows the effect of the hierarchical structure.

Before training, we initialize embeddings with the embeddings generated by TransE, the dimension of the embedding is set to be 50 . We train each subtask separately and limit the training times as 500 for each task, then we pick the optimal parameters for each subtask, respectively.

## Link Prediction

Given $h$ and $r$ in a triple ( $h, r, ?$ ), link prediction aims to rank all the possible tail entities to complete the triple. For each triple, we construct several corresponding "corrupted" triples, which are not in the knowledge graph and considered as negative examples. For a query triple, all entities are ranked in descending order according to the likelihood that they are the answer entity, then the rank of the correct answer entity is recorded for evaluation.

We adopt three metrics for evaluation: mean average precision (MAP), percentage of correct entities ranked in top N (Hits@N), and mean reciprocal rank of correct entities (MRR). The higher these three metrics are, the better the evaluated method is. Results of Hits@N and MRR are shown in Table 2 and the results of MAP are shown in Table 4. Hits@N and MRR are mainly adopted to evaluate embedding-based methods, so they are not employed to evaluate models designed for fact prediction, like DeepPath. We also show the MAP scores for different subtasks on dataset NELL-995 in Table 3.

Compared with baselines, we can see that our method yields competitive results to the state-of-theart models, especially in the case of Hits@N and MRR. This implies that our method can effectively reason on the correct missing entities. Table 3 shows that our approach brings a wide range of improvements in different tasks. Since hierarchical RL is able to model more complicated reasoning processes and has more expressive power compared to single-layer RL, the introduction of a high-level abstraction layer of reasoning improve the accuracy of prediction significantly compared to baselines.

TABLE 2. Link prediction results (hits@n \& MRR) on datasets NELL-995 and FB15k-237.

| Model | NELL-995 |  |  |  | FB15k-237 |  |  |  |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 | MRR |
| TransE | 0.241 | 0.392 | 0.413 | 0.307 | 0.206 | 0.316 | 0.454 | 0.289 |
| TransR | 0.239 | 0.399 | 0.411 | 0.313 | 0.229 | 0.331 | $\mathbf{0 . 5 1 3}$ | 0.291 |
| DistMult | 0.347 | 0.454 | 0.495 | 0.410 | 0.193 | 0.307 | 0.409 | 0.243 |
| CompIEx | 0.382 | 0.473 | 0.522 | 0.467 | 0.204 | 0.316 | 0.420 | 0.261 |
| ConvE | 0.452 | 0.564 | 0.629 | 0.587 | 0.241 | 0.354 | 0.490 | 0.312 |
| MINERVA | 0.663 | 0.773 | 0.831 | 0.725 | 0.217 | 0.329 | 0.456 | 0.293 |
| Ours | $\mathbf{0 . 6 7 9}$ | $\mathbf{0 . 8 0 1}$ | $\mathbf{0 . 8 6 2}$ | $\mathbf{0 . 7 3 4}$ | $\mathbf{0 . 3 2 5}$ | $\mathbf{0 . 4 4 2}$ | 0.511 | $\mathbf{0 . 3 2 6}$ |

TABLE 3. Link prediction results (MAP) on different relations of dataset NELL-995. we report the results of 10 subtasks in detail as in previous work.

| Tasks | TransE | TransR | PRA | DeepPath | MINERVA | AttnPath | Our model |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| athleteHomeStadium | 0.718 | 0.722 | 0.859 | 0.890 | 0.895 | 0.894 | $\mathbf{0 . 9 3 7}$ |
| athletePlaysForTeam | 0.627 | 0.673 | 0.547 | 0.750 | 0.824 | 0.761 | $\mathbf{0 . 8 6 2}$ |
| athletePlaysInLeague | 0.773 | 0.912 | 0.841 | 0.960 | 0.970 | 0.965 | $\mathbf{0 . 9 8 5}$ |
| athletePlaysSport | 0.876 | 0.963 | 0.474 | 0.957 | 0.985 | 0.970 | 0.979 |
| organizationHeadquarteredInCity | 0.620 | 0.657 | 0.811 | 0.790 | 0.946 | 0.941 | $\mathbf{0 . 9 5 4}$ |
| organizationHiredPerson | 0.719 | 0.737 | 0.599 | 0.742 | 0.851 | 0.816 | $\mathbf{0 . 8 8 2}$ |
| personBornInLocation | 0.712 | 0.812 | 0.668 | 0.757 | 0.793 | 0.786 | $\mathbf{0 . 8 2 1}$ |
| personLeadsOrganization | 0.751 | 0.772 | 0.700 | 0.795 | - | 0.828 | $\mathbf{0 . 8 4 0}$ |
| teamPlaysSport | 0.761 | 0.814 | 0.791 | 0.738 | 0.846 | 0.821 | 0.843 |
| worksFor | 0.677 | 0.692 | 0.681 | 0.711 | 0.825 | 0.775 | $\mathbf{0 . 8 3 6}$ |

## Ablation Test

We conducted ablation tests on the structure of the model. The model with single-layer RL and the

TABLE 4. Link prediction results (MAP) on datasets NELL-995 and FB15k-237.

| Models | NELL-995 | FB15k-237 |
| :--- | :---: | :---: |
| TransE | 0.737 | 0.532 |
| TransR | 0.789 | 0.540 |
| PRA | 0.675 | 0.541 |
| RNN-Chain | 0.790 | 0.512 |
| DeepPath | 0.796 | 0.572 |
| MINERVA | - | 0.552 |
| AttnPath | 0.858 | $\mathbf{0 . 6 6 1}$ |
| Ours | $\mathbf{0 . 8 9 4}$ | 0.601 |

model with hierarchical RL but only getting rewards of higher level are set as baselines. Ablation tests' results are shown in Table 5. It confirms the effectiveness of the hierarchical structure and the reward setting.

For hierarchical RL with only high-level reward, consider the simplest case: each step in the high-level

TABLE 5. Ablation test results (MAP) on datasets NELL-995 and FB15k-237.

| Models | NELL-995 | FB15k-237 |
| :--- | :---: | :---: |
| Single-layer RL | - | 0.552 |
| Hierarchical RL <br> with only high- <br> level reward | 0.882 | 0.583 |
| Our model | $\mathbf{0 . 8 9 4}$ | $\mathbf{0 . 6 0 1}$ |

corresponds to only one step in the low level, then the history in high level can be written in form:

$$
\begin{equation*}
\boldsymbol{h}_{t}^{h}=\mathbf{L S T M}\left(\boldsymbol{h}_{t-1}^{h},\left[\operatorname{Re} L U\left(W\left[h_{t-1}^{h} ; \boldsymbol{o}_{t-1}^{h}\right]\right) ; \boldsymbol{o}_{t}^{h}\right]\right) \tag{10}
\end{equation*}
$$

compared to the original form of history in single level RL:

$$
\begin{equation*}
\boldsymbol{h}_{t}=\mathbf{L S T M}\left(\boldsymbol{h}_{t-1},\left[\boldsymbol{a}_{t-1} ; \boldsymbol{o}_{t}\right]\right) \tag{11}
\end{equation*}
$$

a neural network is added into the reasoning, and therefore the expressiveness of the model is improved.

The model in this article goes further to give the lower level reasoning a reward based on embeddings and TransE, which introduces more information into the original reasoning process, so the model performance is further improved.

## CONCLUSION

In this article, we proposed a hierarchical RL-based reasoning model on knowledge graphs. Compared to existing reasoning models based on the single-level RL, we added a high-level abstract conceptual reasoning layer to guide the reasoning on concrete entities at the low level. Our model achieves competitive performance on the link prediction task for two large datasets, and the effectiveness of the proposed hierarchical structure is validated by comparison with MINERVA.

## ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0103405, in part by the National Natural Science Foundation of China under Grant 71621002, and in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27030100.

## REFERENCES

1. T. G. Dietterich, "Hierarchical reinforcement learning with the MAXQ value function decomposition," J. Artif. Intell. Res., vol. 13, pp. 227-303, 2000.
2. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," Proc. 26th Int. Conf. Adv. Neural Inf. Process. Syst., Lake Tahoe, USA, 2013, pp. 2787-2795.
3. Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in Proc. 29th AAAI Conf. Artif. Intell., Austin, TX, USA, 2015, pp. 2181-2187.
4. B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in Proc. Int. Conf. Learn. Representations, USA, 2015.
5. T. Trouillon, J. Welbl, S. Riedel, Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in Proc. Int. Conf. Mach. Learn., vol. 48, Jun. 2016, pp. 2071-2080.
6. N. Lao, T. Mitchell, and W. W. Cohen, "Random walk inference and learning in a large scale knowledge base," in Proc. Conf. Empirical Methods Natural Lang. Process., Edinburgh, Scotland, U.K., Jul. 2011, pp. 529-539.
7. A. Neelakantan, B. Roth, and A. McCallum, "Compositional vector space models for knowledge base completion," in Proc. 53rd Annu. Meet. Assoc. Comput. Linguistics/7th Int. Joint Conf. Asian Federation Nat. Lang. Process., Beijing, China, 2015, pp. 156-166.
8. R. Das, A. Neelakantan, D. Belanger, and A. McCallum, "Chains of reasoning over entities, relations, and text using recurrent neural networks," in Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, Valencia, Spain, Apr. 2017, pp. 132-141.
9. W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A reinforcement learning method for knowledge graph reasoning," in Proc. Empirical Methods Nat. Lang. Process., Copenhagen, Denmark, Sep. 2017, pp. 564-573.
10. R. Das et al., "Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning," in Proc. Int. Conf. Learn. Representations, 2018.
11. H. Wang, S. Li, R. Pan, and M. Mao, "Incorporating graph attention mechanism into knowledge graph reasoning based on deep reinforcement learning," in Proc. Empirical Methods Natural Lang. Process., Hong Kong, China, 2019, pp. 2623-2631.
12. R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," Mach. Learn., vol. 8, no. 3/4, pp. 229-256, May 1992.
13. A. Carlson et al., "Toward an architecture for neverending language learning," in Proc. 24th AAAI Conf. Artif. Intell., Atlanta, GA, USA, 2010, pp. 1306-1313.
14. K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in Proc. Empirical Methods Nat. Lang. Process., Lisbon, Portugal, Sep. 2015, pp. 1499-1509.
15. T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 1811-1818.

ZIKANG WANG is currently working toward the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, China. Her research interests include knowledge graph and natural language processing. She received the B.S. degree in computer science from Central South University, Hunan, China. Contact her at wangzikang2016@ia.ac.cn.

LINJING LI is currently an associate professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include game theory, mechanism design, auction theory, and machine learning. He received the B.E. degree in electrical engineering and automation, and the M.E. degree in control theory and control engineering from the Harbin Institute of Technology, Harbin, China, and the Ph.D. degree in computer applied technology
from the Graduate University of the Chinese Academy of Sciences, Beijing, China. He is a member of the IEEE. Contact him at linjing.li@ia.ac.cn.

DANIEL DAJUN ZENG is a research fellow with the Institute of Automation, Chinese Academy of Sciences. His research interests include intelligence and security informatics, infectious disease informatics, social computing, recommender systems, software agents, and applied operations research and game theory. He received the B.S. degree in economics and operations research from the University of Science and Technology of China, Hefei, China, and the M.S. and Ph.D. degrees in industrial administration from Carnegie Mellon University, Pittsburgh, PA, USA. He is a Fellow of the IEEE. Contact him at dajun.zeng@ia.ac.cn.

## [Professional <br> TECHNOLOGY SOLUTIONS FOR THE ENTERPRISE

## CALL FOR ARTICLES

IT Professional seeks original submissions on technology solutions for the enterprise. Topics include

- emerging technologies,
- social software,
- cloud computing,
- Web 2.0 and services,
- cybersecurity,
- mobile computing,
- green IT,
- RFID,
- systems integration,
- communication networks,
- datacenter operations,
- IT asset management, and
- data management and mining,

We welcome articles accompanied by web-based demos. For more information, see our author guidelines at www.computer.org/itpro/author.htm.


[^0]:    1541-1672 © 2021 IEEE
    Digital Object Identifier 10.1109/MIS.2021.3095055
    Date of publication 8 July 2021; date of current version 11 April 2022.

