

融合 TuckER 嵌入和强化学习的知识推理^①



于铁忠^{1,2}, 罗婧¹, 王利琴^{1,3,4}, 董永峰^{1,3,4}

¹(河北工业大学 人工智能与数据科学学院, 天津 300401)

²(石家庄学院 计算机科学与工程学院, 石家庄 050035)

³(河北省大数据计算重点实验室, 天津 300401)

⁴(河北省数据驱动工业智能工程研究中心, 天津 300401)

通信作者: 王利琴, E-mail: wangliqin@scse.hebut.edu.cn

摘要: 知识推理是补全知识图谱的重要方法, 旨在根据图谱中已有的知识, 推断出未知的事实或关系. 针对多数推理方法仍存在没有充分考虑实体对之间的路径信息, 且推理效率偏低、可解释性差的问题, 提出了将 TuckER 嵌入和强化学习相结合的知识推理方法 TuckRL (TuckER embedding with reinforcement learning). 首先, 通过 TuckER 嵌入将实体和关系映射到低维向量空间, 在知识图谱环境中采用策略引导的强化学习算法对路径推理过程进行建模, 然后在路径游走进行动作选择时引入动作修剪机制减少无效动作的干扰, 并将 LSTM 作为记忆组件保存智能体历史动作轨迹, 促使智能体更准确地选择有效动作, 通过与知识图谱的交互完成知识推理. 在 3 个主流大规模数据集上进行了实验, 结果表明 TuckRL 优于现有的大多数推理方法, 说明将嵌入和强化学习相结合的方法用于知识推理的有效性.

关键词: 知识图谱; 知识推理; TuckER 嵌入; 强化学习; 路径搜索; 路径规划

引用格式: 于铁忠, 罗婧, 王利琴, 董永峰. 融合 TuckER 嵌入和强化学习的知识推理. 计算机系统应用, 2022, 31(9): 127-135. <http://www.c-s-a.org.cn/1003-3254/8681.html>

Knowledge Reasoning Combining TuckER Embedding and Reinforcement Learning

YU Tie-Zhong^{1,2}, LUO Jing¹, WANG Li-Qin^{1,3,4}, DONG Yong-Feng^{1,3,4}

¹(School of Artificial Intelligence and Data Science, Hebei University of Technology, Tianjin 300401, China)

²(School of Computer Science and Engineering, Shijiazhuang University, Shijiazhuang 050035, China)

³(Hebei Key Laboratory of Big Data Computing, Tianjin 300401, China)

⁴(Hebei Engineering Research Center of Data-driven Industrial Intelligent, Tianjin 300401, China)

Abstract: Knowledge reasoning is an important method to complement a knowledge graph, which aims to infer unknown facts or relations according to the existing knowledge in the graph. As the path information between entity pairs is not fully considered in most reasoning methods, the reasoning shows low efficiency and poor interpretability. To solve this problem, this study proposes TuckER embedding with reinforcement learning (TuckRL), a knowledge reasoning method that combines TuckER embedding and reinforcement learning (RL). First, entities and relations are mapped to low-dimensional vector space through TuckER embedding, and the path reasoning process is modeled using RL guided by strategies in the knowledge graph environment. Then, the action pruning mechanism is introduced to reduce the interference of invalid actions for action selection during path walking, and LSTM is used as the memory component to preserve the agent's historical action trajectory. In this way, the agent can more accurately select valid actions and can complete knowledge reasoning by interaction with the knowledge graph. The experiments on three mainstream large-scale

① 基金项目: 国家自然科学基金 (61806072); 天津市自然科学基金 (19JCZDJC40000); 河北省高等学校科学技术研究项目 (QN2021213); 河北省自然科学基金 (F2020202008)

收稿时间: 2021-12-16; 修改时间: 2022-01-18; 采用时间: 2022-01-26; csa 在线出版时间: 2022-06-28

datasets indicate that TuckRL is superior to most of the existing methods, which demonstrates the effectiveness of combining embedding and RL for knowledge reasoning.

Key words: knowledge graph (KG); knowledge reasoning; TuckER embedding; reinforcement learning (RL); path search; path planning

1 引言

知识图谱 (knowledge graph, KG)^[1]本质上是一种概念网络,其基本组成单位是形式为(实体,关系,实体)的三元组,目前已经构建了许多知识图谱,如 WordNet^[2]、NELL^[3]、Freebase^[4]等,并成功应用于信息检索、推荐系统、问答系统等智能服务领域.因为构建的大规模知识图谱通常是不完备的,需要不断对其进行补充,而知识推理^[5,6]是从现有的数据中推理出新的实体或关系,从而不断完善知识图谱,因此知识推理近年来成为知识图谱研究领域的热点问题之一.

使用张量因子分解或神经网络学习实体和关系的嵌入是目前较为流行的知识推理方法^[7-9],它们将知识图谱中的实体和关系表示成低维稠密向量,然后利用向量的相似性推理出实体之间的关系或者判定给定的三元组是否为真,从而补全知识图谱.这些方法效率较高,但是依赖于知识图谱的三元组表示形式,大多数都没有捕捉到多跳路径的关系,从而限制了其在更复杂的推理任务中的应用.因此,结合实体对之间的多跳路径信息成为知识推理的另一种解决方案.路径排序算法 (path ranking algorithm, PRA)^[10]使用基于重启推理机制的随机游走来执行多个有界深度优先搜索,通过监督学习选择更合理的路径,在一定程度上解决了上述问题.由于具有可解释性和良好的性能,最近的工作将多跳推理表述为马尔可夫决策过程 (Markov decision process, MDP)^[11],并利用强化学习 (reinforcement learning, RL)^[12,13]执行有效路径搜索.其中 DeepPath^[14]是第一个将强化学习迁移到知识图谱中的多跳推理方法,该方法将实体对应强化学习中的状态,关系对应动作,旨在使用 RL 对关系进行采样来扩展路径,这类方法的提出为知识图谱的推理提供了新的研究思路.

然而在实际知识推理任务当中,使用强化学习来进行路径搜索的效率并不高,一方面,多数强化学习方法在构建知识图谱环境时没有对实体和关系进行较好的嵌入,从而导致智能体的路径搜索成功率偏低;另一方面,知识图谱中存在大量无效路径,比如对于三元组

(人物 A , 出生于, 北京)、(北京, 位于, 中国)、(人物 A , 结婚, 人物 B)而言,可以推出(人物 B , 出生于, 中国).在推理过程中,关系“结婚”是实体,“北京”和“中国”的无效动作,因为实体“北京”和“中国”不能作为“结婚”的主语或者宾语.当智能体在游走的过程中选择了某无效动作时,会停止并退回上一步重新进行选择,然而智能体可能会不断选择该无效动作;当选择有效路径时,也会存在同一条路径重复被选择的情况,均会造成游走死循环.上述情况都有可能导导致智能体在游走初始阶段难以获得策略网络给予的奖励,使得路径选择的准确率降低.

针对以上提出的在进行知识推理时路径选择效率和准确率偏低的问题,本文实现了表示学习和强化学习的融合,提出一种融合 TuckER 嵌入和强化学习的知识图谱推理方法 TuckRL (TuckER reinforcement learning, TuckRL),将路径选择问题转化为序列决策问题.通过使用 TuckER 嵌入得到知识图谱中实体和关系的向量表示,使智能体在与知识图谱环境的交互中能够更精准的搜索路径,提高了推理方法的效率;为了减少无效动作对智能体的干扰,并鼓励策略网络选择不同的关系,通过对动作执行随机丢弃操作,使智能体得到更全面的训练,提高了模型的泛化能力;使用长短期记忆网络 (long short-term memory, LSTM) 存储智能体历史动作,在关系选择时强制智能体选择其他动作来避免在同一实体节点上不断停顿,使其在训练过程中尽可能为推理任务找到成功率较高的路径.

2 相关工作

目前面向知识图谱的知识推理方法可分为以下3类:基于嵌入的方法、基于关系路径的方法以及基于强化学习的方法,本节将按照此分类对知识推理方法的国内外研究工作进行概述.

基于嵌入的推理是将实体和关系映射到向量空间中,通过计算得到的向量相似度完成推理.其中最经典的模型是 Bordes 等^[8]于2013年提出来的 TransE,该

模型将关系视为实体对之间的某种翻译,在处理简单关系时表现良好,但在面对 1- N 、 N -1、 N - N 等复杂关系时会存在错误;Wang 等^[15]提出的 TransH 通过设置一个关系超平面,使不同关系下的实体有不同的表示,解决了 TransE 在处理复杂关系时的局限性。Ji 等^[16]提出的 TransD 同时考虑实体和关系的多样性,通过设置两个投影矩阵分别将头尾实体投影到关系空间,模型更加灵活;陈文杰等^[17]提出的 TransGraph 模型学习三元组信息的同时,还考虑到知识图谱的网络结构特征和语义信息,从而增强三元组的表示效果;Trouillon 等^[18]提出的 ComplEx 将实体和关系嵌入到复数向量空间中,以推理对称和反对称关系;Balažević等^[19]提出的 TuckER 将知识图谱表示成三阶二元张量,每个元素对应一个三元组,具有较强的学习特征能力。

基于关系路径的推理方法侧重于捕捉 KG 中路径上的信息,也就是说,此类方法不仅可以预测实体之间的直接关系,还考虑其多跳关系的丰富语义。早期的研究路径排序算法 PRA 通过在 KG 上随机游走得到实体对之间的所有路径,并利用二分类器推理缺失的关系。Lin 等^[20]提出的 PTransE 通过组合每条路径中的所有关系得到路径的嵌入,并设计了路径约束资源分配算法 (path-constraint resource allocation, PCRA) 衡量关系路径的可靠性。Das 等^[21]提出 Path-RNN (path-recurrent neural network, Path-RNN) 神经网络模型,将每条路径分解为关系序列,通过 RNN 组合关系路径的语义信息,构造出路径的向量表示。Wang 等^[22]提出知识感知的路径循环网络 (knowledge-aware path recurrent network, KPRN) 模型,在嵌入实体和向量之后,LSTM 通过组合实体和关系的语义生成路径表示,利用路径中的序列依赖项进行关系补充。

基于强化学习的推理方法将实体之间的路径游走视为马尔可夫决策过程,使用基于策略的智能体搜索推理路径。Xiong 等^[14]提出第一个考虑在知识图谱中学习路径的强化学习方法 DeepPath;Das 等^[23]提出的 MINERVA 是使用强化学习训练用于多跳 KG 查询应答的端到端模型;Shen 等^[24]提出的 M-Walk 是一个由 RNN 和蒙特卡洛树组成的智能体,用来编码路径状态以及生成有效路径。Li 等^[25]提出的 DIVINE 是一种基于生成对抗学习的框架,通过学习推理策略和奖励函数来增强知识图谱中基于 RL 的推理。Meilicke 等^[26]提出基于规则的多跳推理模型,引入强化学习指导规

则采样过程,有助于获取更有价值的规则。Lei 等^[27]提出的 RuleGuider 利用基于符号方法生成的高质量规则为智能体提供奖励监督。崔员宁等^[28]提出的 TransPath 通过在目标任务之外增加单步游走选择有效动作的源任务来提高路径选择的成功率。

从以上研究可以发现,很少有将嵌入模型和强化学习相结合的方法,且在使用强化学习进行推理的过程中,没有充分利用智能体的历史路径信息。因此,本文提出采用 TuckRL 方法解决此类问题,首先将实体和关系进行低维嵌入,完成知识图谱环境的创建;然后在强化学习的框架中,为减少无效动作对推理结果的干扰,通过随机丢弃智能体的输出边进行动作修剪,并引入 LSTM 网络作为记忆组件存储历史路径,以提高路径选择的效率。

3 融合 TuckER 嵌入和强化学习的知识推理

知识推理的具体任务是在实体对之间找到可靠的预测路径,所以为了提高路径搜索的效率和质量,本文提出了一种融合 TuckER 嵌入和强化学习的知识推理方法 TuckRL,将寻找实体对之间可能存在的关系以及路径信息转化为强化学习的序列决策问题。模型如图 1 所示,分为 3 部分。

其中,第 1 部分为知识图谱环境模块:使用 TuckER 嵌入将实体和关系映射成含有三元组语义信息的向量;第 2 部分为强化学习环境模块:将 TuckER 嵌入得到的实体和关系的连续向量化表示作为基于强化学习的神经网络的输入,使得模型能够充分利用知识图谱已经存在的三元组信息,且 RL 智能体在游走的过程中进行策略网络的训练,使用动作修剪和 LSTM 网络来控制关系选择和路径搜索,帮助智能体选择有效动作,以便提高模型的性能;第 3 部分为策略引导的路径推理模块:智能体与知识图谱环境进行交互,在知识推理阶段采用训练好的策略,完成推理任务。

3.1 TuckER 嵌入

知识图谱嵌入是 KG 建模的方法,通过学习评分函数 $f(e_H, e_T)$ 来定义空间中的三元组,使得语义相近的实体在嵌入空间中的向量表示距离也相近。TuckER 具备完全表达能力,即通过训练学习,能够将正三元组和负三元组完全区分开,且性能较优于当前的线性嵌入模型。图 2 为 TuckER 嵌入模型的可视化表示。

在该嵌入模型中,知识图谱被表示为一个三阶二

元张量, 而 Tucker 分解的核心思想是将该三阶张量分解为 1 个核心张量和 3 个矩阵, 每一个元素表示一条事实三元组, 值为 1 表示真实三元组, 为 0 表示错误或缺失事实. 定义一个原始张量 $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, 通过 Tucker 可以分解为核心张量 $\mathcal{Z} \in \mathbb{R}^{P \times Q \times R}$ 和 3 个矩阵 $A \in \mathbb{R}^{I \times P}$ 、 $B \in \mathbb{R}^{J \times Q}$ 、 $C \in \mathbb{R}^{K \times R}$, 如式 (1) 所示:

$$\mathcal{X} \approx \mathcal{Z} \times_1 A \times_2 B \times_3 C \quad (1)$$

其中, \times_n 表示沿第 n 阶模的张量积, 3 个矩阵每一行分别为头实体 e_H 、关系 r 和尾实体 e_T 的向量表示, 而核心

张量 \mathcal{Z} 表征了它们之间的交互级别. 头实体和尾实体是等价的, 均用实体嵌入矩阵 E 来表示, 即 $E = A = C \in \mathbb{R}^{n_e \times d_e}$, 且关系矩阵嵌入为 $R = B \in \mathbb{R}^{n_r \times d_r}$, 其中 n_e 和 n_r 表示实体和关系的数量, d_e 和 d_r 表示实体和关系嵌入向量的维数.

综上, 定义出 Tucker 的得分函数如式 (2) 所示:

$$f(e_H, e_T) = \mathcal{W} \times_1 e_H \times_2 w_r \times_3 e_T \quad (2)$$

其中, $\mathcal{W} \in \mathbb{R}^{d_e \times d_r \times d_e}$ 为核心张量, 即模型参数; $w_r \in \mathbb{R}^{d_r}$ 为 R 的关系表示.

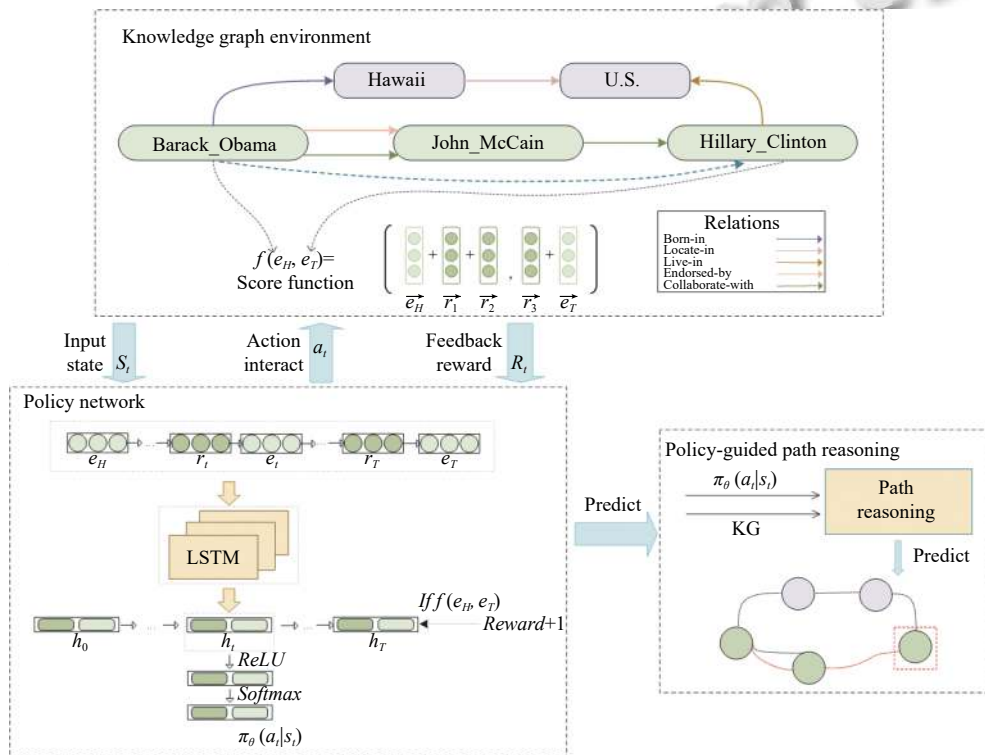


图 1 融合 Tucker 嵌入和强化学习的知识推理模型框架图

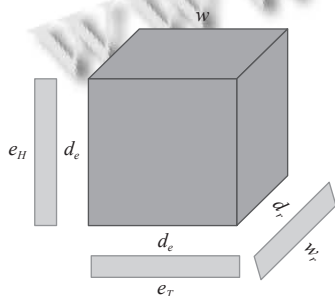


图 2 Tucker 嵌入模型图

3.2 强化学习

TuckRL 模型是将知识图谱推理问题转化为马尔

科夫决策问题, 智能体的状态转移和动作选择都在知识图谱环境中完成, 故本节介绍将知识图谱 \mathcal{G} 建模为强化学习智能体决策环境的过程.

该过程主要由 $\langle S, A, \gamma, P \rangle$ 四部分构成, 其中 S 表示智能体的连续状态空间, $A = \{a_1, a_2, \dots, a_n\}$ 是动作空间, 表示所有可用动作的集合, $\gamma(s, a)$ 为奖励函数, P 是状态转移策略.

(1) 状态空间

知识图谱中的实体集合 E 作为智能体的状态 S , 将智能体在每个时间步骤的状态表示成 $s_t \in S$, 且 e_t 表示第 t 步访问的实体. 为了更好地表达其语义内涵, 采

用 TuckER 嵌入将实体表示成低维稠密向量.

$$s_t = \text{TuckER}(e_t) \quad (3)$$

(2) 动作空间

动作空间 A 被定义为 KG 中的关系集合 \mathcal{R} , A_t 表示状态 s_t 所对应的实体 e_t 在 KG 中所有可能的输出边, 智能体要选择一个边进行路径搜索, 游走步数 T 作为路径搜索的终止条件. 动作集合表示如下:

$$A_t = \{(r', e') | (e_t, r', e') \in \mathcal{G}\} \quad (4)$$

其中, r' 、 e' 分别表示下一步有可能选择的关系和实体.

(3) 状态转移

状态转移是指智能体根据当前状态做出动作移动到下一个状态的过程. 具体来说, 智能体在当前状态下, 通过选择某动作后, 并基于环境的交互实现从当前状态到下一状态的转移. 状态转移 P 表示如下:

$$P = (S_{t+1} = s' | S_t = s, A_t = a) \quad (5)$$

(4) 奖励函数

当智能体从初始状态开始搜索, 最终能够到达目标状态, 则获得正向奖励 1, 否则无奖励. 智能体会根据奖励及时更新自己的策略, 尽可能实现奖励最大化, 奖励函数定义如下:

$$\text{Reward}(s_t) = \begin{cases} 1, & \text{if } (e_H, r_q, e_T) \in \mathcal{G} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

其中, r_q 表示查询到的关系, e_T 表示最终的尾实体.

3.3 基于 MDP 的策略网络

基于 MDP 的策略网络是 TuckRL 的关键部分, 主要用于引导智能体在知识图谱和强化学习的交互环境中进行游走, 做出高效且准确的决策. 在该策略网络中, 首先使用 Dropout 动作修剪, 目的是减少无效动作对于智能体游走的干扰, 提高动作的选择效率, 然后引入长短期记忆网络 LSTM 作为记忆组件, 避免智能体在同一实体节点上不断停滞的同时, 编码完整的路径游走历史轨迹, 并采用策略梯度下降算法更新策略网络的参数, 以便在推理过程中引导智能体走向更可靠的路径.

(1) 动作修剪

由于无效路径通常比正确路径多, 且更容易被发现, 从而增加了路径搜索的负担, 尤其是当 KG 随着路径跳数的增长, 动作空间也会呈指数型增加, 从而加大搜索负担, 对于出度较大的实体 (即与之相连的关系较多), 这种现象会更严重. 而枚举出来实体对之间所有可能的关系路径在大型知识图谱上是不可行的, 因此

如何进行有效的路径探索, 找出推理路径格外重要.

针对这类问题, TuckRL 借鉴深度神经网络中 Dropout 丢弃神经元来缓解过拟合的思想, 提出了一种新的训练机制: 将随机丢弃思想用到强化学习的路径选择中, 按照一定的概率屏蔽当前实体的输出边, 从而实现动作的修剪. 在智能体采样路径的时候, 随机屏蔽当前状态的一些输出边, 智能体根据修剪后的关系分布来采样动作. 这样不仅可以减少智能体的搜索空间, 防止 GPU 内存溢出, 同时还能进一步扩大对于不同路径集的有效探索, 提高路径选择的随机性.

(2) LSTM 编码路径

KG 中的每一个实体和关系都通过 TuckER 嵌入分别得到具有语义信息的低维稠密向量 $e \in E$, $r \in R$, 为保存搜索历史路径, 利用 3 层 LSTM 记忆和编码智能体的历史动作. 假设初始状态为 $h_0 = 0$, 搜索历史 $h_t = (e_H, r_1, e_1, \dots, r_t, e_t)$ 由从第 1 步到第 t 步所采取的动作序列构成.

$$h_0 = \text{LSTM}(0, [r_0; e_H]) \quad (7)$$

$$h_t = \text{LSTM}(h_{t-1}, a_{t-1}), t > 1 \quad (8)$$

其中, r_0 表示初始关系, e_H 为初始实体.

(3) 策略神经网络优化

在知识推理中, 基于策略网络的强化学习将输入状态 s_t 映射到所有可能被选择的动作的概率向量中, 本文将组合得到的状态向量输入到由两个隐藏层组成的神经网络中, 且每个隐藏层后面会有一个 ReLU 层, 输出层使用 Softmax 进行归一化. 策略网络 π 定义如下:

$$\pi_\theta(a_t | s_t) = \text{Softmax}(A_t \times W \text{ReLU}[h_t, (s_t, a_t)]) \quad (9)$$

其中, θ 为神经网络参数, 动作空间 A_t 为所有动作嵌入的集合, W 为隐藏层的权重.

对于上述策略网络 π_θ , 模型使用 REINFORCE 梯度策略方法来优化参数 θ , 如式 (10) 所示:

$$J(\theta) = \mathbb{E}_{(e_H, r, e_t) \in \mathcal{G}} [\mathbb{E}_{a_1, \dots, a_T \sim \pi_\theta} [\text{Reward}(s_T | e_H, r_t)]] \quad (10)$$

其中, $J(\theta)$ 表示一个批次的奖励, \mathbb{E} 表示训练集上不同三元组对应的期望值.

REINFORCE 梯度策略方法使用当前的策略生成的一系列历史轨迹 (迭代遍历所有在 \mathcal{G} 中的三元组) 来估计随机梯度, 然后用随机梯度来更新参数, 如式 (11) 和式 (12) 所示:

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_{t=1}^T \gamma(s_t | e_H, r_t) \log \pi_\theta(a_t | s_t) \quad (11)$$

$$\theta = \theta + \beta \nabla_{\theta} J(\theta) \quad (12)$$

其中, θ 为需要更新的参数, e_H 表示头实体, r_t 表示当前关系, $\pi_{\theta}(a_t|s_t)$ 为在 s_t 状态下策略网络选择 a_t 的概率. γ 为执行该动作所获得的奖励值, β 为学习率.

3.4 模型训练

在进行知识推理之前, 对 RL 智能体进行策略网络的训练, 目的是让智能体在路径游走的过程中尽可能地直接选择正确的动作, 从而更高效的完成多步关系推理. 在训练中, 首先输入知识图谱训练集 *Train*、限制游走长度的最大步数 T , 然后是智能体和图谱环境之间的迭代: 根据策略网络的输出结果, 选择一个关系 r 作为下一步的执行动作, 此时判断起始状态 s_0 和目前选择的关系 r 组成的三元组是否在知识图谱 \mathcal{G} 中, 若是, 给予奖励并更新策略网络, 具体训练过程如算法 1.

算法 1. 策略网络训练算法

输入: 知识图谱 \mathcal{G} , 训练集 *Train*, 最大步数大小 T

输出: 策略网络 π_{θ} 的参数 θ

```

1) for episode 1 to  $N$  do
2)   initial  $s_t = s_0, h_0 = 0, step = 0$ 
3)   while  $step < T$  do
4)      $s_t = \text{TuckER}(e_t), a_t = \text{TuckER}(r_t)$ 
5)     Update  $h_t = \text{LSTM}(h_{t-1}, a_{t-1})$  //根据策略  $\pi_{\theta}(a_t|s_t)$  执行动作  $a$ , 得到奖励  $Reward$  和状态  $S_{t+1}$ 
6)     if  $Reward = 0$ 
7)        $step++$ 
8)     if  $Reward = 1$  or  $step = T$ 
9)       break
10)    end while
11)   $g \propto \nabla_{\theta} \sum_{i=1}^T Reward(s_i|e_s, r) \log \pi_{\theta}(a_i|s_i)$  //更新策略网络
12) end for

```

4 实验分析

4.1 数据集及评价标准

为了评估本文所提方法, 在 NELL-995、WN18RR、FB15K-237 这 3 个大规模数据集上进行实验. 其中 NELL-995 是基于语义机器学习系统 NELL 的第 995 次迭代产生的数据集, 包含 7.5k 个实体、200 个关系以及 16.9k 个三元组, WN18RR 的三元组数据来自大型英文语义知识库 WordNet, 包含 4.0k 个实体、11 个关系和 14.6k 个三元组; FB15K-237 是包含常见信息的世界知识库 FB15K 的子集, 通过从验证集和测试集中移除许多关系的逆关系构建, 包含 1.4k 个实体、237 个关系和 29.2k 个三元组; 数据集的统计信息如表 1 所示.

表 1 数据集的信息

数据集	实体数	关系数	训练集	测试集	平均度数
NELL-995	75 492	200	154 213	15 838	4.07
WN18RR	40 945	11	141 442	5 000	2.19
FB15K-237	14 541	237	272 115	20 466	19.74

在关系预测任务中, 通常使用平均倒数排名 (mean reciprocal rank, *MRR*) 和推理结果命中率 *Hits@N* 作为评估指标. 对于三元组中缺失的关系, 模型会对测试集中的实体对 (e_1, e_2) , 依据评分函数预测出带有顺序的关系集合 $r = \{r_1, r_2, \dots, r_n\}$, 正确的关系 r 在关系集合中排名越靠前, 则说明模型的预测效果越好.

MRR 是指正确结果在所有预测结果中排名的倒数平均值, 计算如式 (13) 所示:

$$MRR = \frac{1}{N} \sum_{i=1}^N \frac{1}{rank_i}, 1 \leq i \leq N \quad (13)$$

Hits@N 表示正确的结果在所有预测结果中排在前 n 位所占的比例, 计算如式 (14) 所示:

$$Hits@N = \frac{1}{N} \sum_{i=1}^N I, I = \begin{cases} 1, & \text{if } rank_i \leq n \\ 0, & \text{if } rank_i > n \end{cases} \quad (14)$$

其中, N 是需预测关系的实体对数量; $rank_i$ 是对需预测的第 i 个实体对而言正确的关系在所有预测结果中的排序位置. I 为指示函数, 表示当 $rank_i \leq n$ 时, $I = 1$, 否则 $I = 0$.

4.2 参数设置

实验使用与 RuleGuider^[27] 相同的训练集、验证集和测试集. 对于每个实体, 将实体的最大输出边数设置为阈值 η , 以防止 GPU 内存溢出, 并保留其具有最高 PageRank 分数的前 n 个邻居. 将 TuckER 模型嵌入得到的所有实体和关系向量维度设置为 100, 这也是策略网络 Policy Network 的输入大小, 路径编码器 LSTM 的隐藏层维度设置为 100. 选择 Adam 作为优化器, 学习率 β 分别设置为 $\{0.001, 0.002, 0.003\}$, 对于整个训练过程, Dropout 分别为 $\{0, 0.1, 0.2, 0.3, 0.4\}$, 迭代次数 $num_epoches$ 和批大小 $batch_size$ 分别为 20 和 128.

4.3 实验结果与分析

为了验证所提方法的性能, 与嵌入模型 TransE、DistMult、ComplEx 和 ConVKB^[29], 使用强化学习进行推理的模型 DeepPath、MINERVA、AnyBURL 和 RuleGuider 进行对比实验, 得出本模型 TuckRL 优于大部分模型, 实验结果如表 2 所示, 其他模型的实验结果

使用文献 [27] 给出的结果. 由表 2 可知, 基于嵌入的模型虽然比较简单, 但是在多个数据集上的整体结果是不错的, 原因可能是基于嵌入的方法可以将 KG 中的每个三元组映射到嵌入空间, 从而可以编码整个图谱的连通性; 而 TuckRL 也正是利用了嵌入模型的这个优点, 也得到了不错的实验结果. 在 WN18RR 和 FB15K-237 数据集上比其他方法有较明显提升, 尤其在 FB15K-

237 数据集上, 主要原因可能是 FB15K-237 实体之间路径长度较长, 在其他模型中动作选择的正确率较低, 而 TuckRL 中使用动作修剪和 LSTM 编码路径有效避免了选择无效动作. 虽然在大型数据集 NELL-995 上本文的方法相对于 MINERVA 没有得到显著改善, 但 $Hits@1$ 、 $Hits@3$ 和 MRR 指标略优于最新模型 RuleGuider 和多数嵌入模型.

表 2 不同推理方法在不同数据集上的命中率实验结果分析比较 (%)

Model	NELL-995				WN18RR				FB15K-237			
	$Hits@1$	$Hits@3$	$Hits@10$	MRR	$Hits@1$	$Hits@3$	$Hits@10$	MRR	$Hits@1$	$Hits@3$	$Hits@10$	MRR
TransE	40.1	—	35.2	34.4	28.9	47.5	<u>56.0</u>	35.9	27.8	19.8	37.6	44.1
DistMult	55.2	—	78.3	64.1	41.0	—	47.5	43.3	32.4	—	<u>60.0</u>	41.7
ComplEx	63.9	—	84.8	72.1	41.5	—	46.9	43.4	<u>32.8</u>	—	61.6	<u>43.2</u>
ConvKB	47.0	57.0	64.5	53.1	15.8	49.5	55.8	36.5	29.8	32.4	47.1	38.9
DeepPath	52.6	61.5	77.3	63.8	35.3	36.7	42.8	39.2	16.9	24.8	35.7	22.7
MINERVA	66.3	<u>77.3</u>	83.1	72.5	41.3	45.6	51.3	44.8	21.7	32.9	45.6	29.3
AnyBURL	44.0	53.9	57.0	—	<u>42.9</u>	<u>49.6</u>	53.7	—	26.9	<u>38.4</u>	52.0	—
RuleGuider	65.0	—	83.1	<u>72.9</u>	42.2	—	53.4	<u>44.9</u>	31.6	—	55.1	42.1
TuckRL	<u>65.8</u>	78.5	<u>83.5</u>	73.1	43.5	54.4	57.3	47.7	36.9	46.3	59.1	42.6

注: 加粗字体表示最优实验结果, 下划线表示次优实验结果.

4.4 Dropout 分析

为了考察动作修剪策略对于模型性能的影响, 在 FB15K-237 引入 Dropout={0, 0.1, 0.2, 0.3, 0.4} 进行实验, 命中率 $Hits@N$ 和平均倒数排名 MRR 的实验结果如图 3 所示, 可以观察到, $Hits@N$ 和 MRR 一开始随着 Dropout 的增加而得到提升. 在 Dropout=0.3 达到最高值, 并之后开始有所下降, 尤其在 $Hits@10$ 评价指标上最为明显. 分析可知, 一开始 Dropout 实现了动作修剪, 减少了动作空间的大小, 提高了路径选择的正确性和效率, 但是随着 Dropout 的增加, 有一些有效动作也会伴随着大量的舍弃, 从而导致结果的降低.

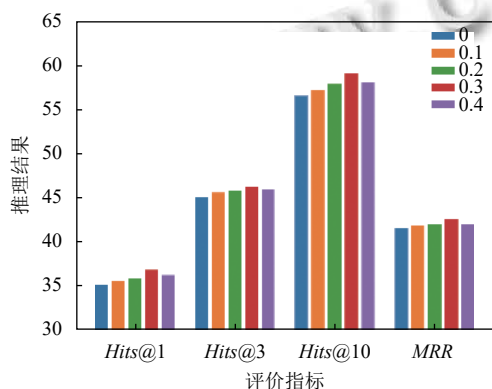


图 3 在 FB15K-237 数据集上不同 Dropout 率的实验结果

4.5 消融实验

为了研究本文模型 TuckRL 中各个组件的重要性,

在 NELL-995、WN18RR 和 FB15K-237 数据集上通过替换 TuckER 嵌入方法 (-TuckER)、移除 Dropout (-Dropout) 和移除路径编码器 LSTM (-LSTM) 进行消融实验的研究, 并将最终的 $Hits@3$ 和 MRR 结果与整个模型进行了比较, 实验结果如表 3 所示. 由表 3 可知, 移除每个组件都会导致模型性能的下降, 每个组件对模型的最终结果都有不同的影响.

表 3 消融实验结果

Model	NELL-995		WN18RR		FB15K-237	
	$Hits@3$	MRR	$Hits@3$	MRR	$Hits@3$	MRR
-TuckER	72.9	70.3	46.2	44.9	40.1	39.8
-Dropout	76.3	71.9	52.6	46.3	45.1	41.6
-LSTM	69.6	68.4	49.5	40.2	37.7	36.5
TuckRL	78.5	73.1	54.4	47.7	46.3	42.6

在该消融实验中, 首先将 TuckER 嵌入替换为 ComplEx 嵌入, 原始结果在 NELL-995、FB15K-237 和 WN18RR 数据集上的 $Hits@3$ 和 MRR 分别有不同程度的下降, 由此可见, 一个性能较好的嵌入方法对于知识推理的作用也是较为明显的; 当移除 Dropout 动作丢弃组件时, 发现对应的结果也均得到了降低, 由此可见, 删除动作丢弃组件会对策略网络中路径的游走有一定的影响; 最后是去掉可以记忆历史路径的 LSTM 组件, 也就是说, 智能体仅根据当前的实体和策略函数来选择下一步的动作, 并且无法得到历史路径 h_t , 从实

验结果可以观察到记忆组件 LSTM 对于模型的性能具有重要影响。

5 结论与展望

本文提出了一种基于强化学习的知识图谱推理方法 TuckRL, 该方法融合了表示学习和强化学习, 设计出一个全新的路径游走策略, 在强化学习智能体进行关系选择和路径游走的过程中, 引入了 Dropout 动作修剪机制和 LSTM 神经网络, 相比之前的强化学习工作, 更有利于智能体在推理过程中更有效的挖掘高质量路径, 从而完成知识图谱推理任务. 实验部分验证了本文模型的性能, 但是基于 RL 的方法与基于嵌入的方法在个别指标上的差距仍然存在. 在未来的研究中, 将引入注意力机制来对邻居节点分配不同的权重, 从而更好地捕获两个实体之间每条路径的语义相关性, 来得到更好的推理效果.

参考文献

- 1 黄恒琪, 于娟, 廖晓, 等. 知识图谱研究综述. 计算机系统应用, 2019, 28(6): 1–12. [doi: 10.15888/j.cnki.csa.006915]
- 2 Miller GA. WordNet: A lexical database for English. Communications of the ACM, 1995, 38(11): 39–41. [doi: 10.1145/219717.219748]
- 3 Carlson A, Betteridge J, Kisiel B, *et al.* Toward an architecture for never-ending language learning. Proceedings of the 24th AAAI Conference on Artificial Intelligence. Menlo Park: AAAI, 2010. 1306–1313.
- 4 Bollacker KD, Evans C, Paritosh P, *et al.* Freebase: A collaboratively created graph database for structuring human knowledge. Vancouver: ACM, 2008. 1247–1250.
- 5 封皓君, 段立, 张碧莹. 面向知识图谱的知识推理综述. 计算机系统应用, 2021, 30(10): 21–30. [doi: 10.15888/j.cnki.csa.008137]
- 6 官赛萍, 靳小龙, 贾岩涛, 等. 面向知识图谱的知识推理研究进展. 软件学报, 2018, 29(10): 2966–2994. [doi: 10.13328/j.cnki.jos.005551]
- 7 Yang BS, Yih WT, He XD, *et al.* Embedding entities and relations for learning and inference in knowledge bases. arXiv: 1412.6575, 2014.
- 8 Bordes A, Usunier N, Garcia-Durán A, *et al.* Translating embeddings for modeling multi-relational data. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2013. 2787–2795.
- 9 Dettmers T, Minervini P, Stenetorp P, *et al.* Convolutional 2D knowledge graph embeddings. Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans: AAAI, 2018. 1811–1818.
- 10 Lao N, Mitchell TM, Cohen WW. Random walk inference and learning in a large scale knowledge base. Proceedings of the Conference on Empirical Methods in Natural Language Processing. Edinburgh: ACM, 2011. 529–539.
- 11 Bellman R. A Markovian decision process. Journal of Mathematics and Mechanics, 1957, 6(5): 679–684.
- 12 Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 1992, 8(3–4): 229–256. [doi: 10.1007/BF00992696]
- 13 李茹杨, 彭慧民, 李仁刚, 等. 强化学习算法与应用综述. 计算机系统应用, 2020, 29(12): 13–25. [doi: 10.15888/j.cnki.csa.007701]
- 14 Xiong WH, Hoang T, Wang WY. DeepPath: A reinforcement learning method for knowledge graph reasoning. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. 564–573.
- 15 Wang Z, Zhang JW, Feng JL, *et al.* Knowledge graph embedding by translating on hyperplanes. Proceedings of the 28th AAAI Conference on Artificial Intelligence. Quebec City: AAAI, 2014. 1112–1119.
- 16 Ji GL, He SZ, Xu LH, *et al.* Knowledge graph embedding via dynamic mapping matrix. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing: Association for Computational Linguistics, 2015. 687–696.
- 17 陈文杰, 文奕, 张鑫, 等. 一种改进的基于 TransE 知识图谱表示方法. 计算机工程, 2020, 46(5): 63–69, 77.
- 18 Trouillon T, Welbl J, Riedel S, *et al.* Complex embeddings for simple link prediction. Proceedings of the 33rd International Conference on Machine Learning. New York: PMLR, 2016. 2071–2080.
- 19 Balažević I, Allen C, Hospedales TM. TuckER: Tensor factorization for knowledge graph completion. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019. 5185–5194.
- 20 Lin YK, Liu ZY, Luan HB, *et al.* Modeling relation paths for representation learning of knowledge bases. Proceedings of

- the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics, 2015. 705–714.
- 21 Das R, Neelakantan A, Belanger D, *et al.* Chains of reasoning over entities, relations, and text using recurrent neural networks. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. Valencia: Association for Computational Linguistics, 2017. 132–141.
- 22 Wang X, Wang DX, Xu CR, *et al.* Explainable reasoning over knowledge graphs for recommendation. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019. 5329–5336.
- 23 Das R, Dhuliawala S, Zaheer M, *et al.* Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. 6th International Conference on Learning Representations. Vancouver: OpenReview.net, 2018.
- 24 Shen YL, Chen JS, Huang PS, *et al.* M-Walk: Learning to walk over graphs using Monte Carlo tree search. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montreal: Curran Associates Inc., 2018. 6787–6798.
- 25 Li RP, Cheng X. DIVINE: A generative adversarial imitation learning framework for knowledge graph reasoning. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019. 2642–2651.
- 26 Meilicke C, Chekol MW, Fink M, *et al.* Reinforced anytime bottom up rule learning for knowledge graph completion. arXiv: 2004.04412, 2000.
- 27 Lei DR, Jiang GR, Gu XT, *et al.* Learning collaborative agents with rule guidance for knowledge graph reasoning. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2020. 8541–8547.
- 28 崔员宁, 李静, 陈琰, 等. TransPath: 一种基于深度迁移强化学习的知识推理方法. 小型微型计算机系统, 2022, 43(3): 536–543.
- 29 Nguyen DQ, Nguyen TD, Nguyen DQ, *et al.* A novel embedding model for knowledge base completion based on convolutional neural network. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans: Association for Computational Linguistics, 2018. 327–333.

(校对责编: 牛欣悦)