软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2023,34(10):4565–4583 [doi: 10.13328/j.cnki.jos.006889]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向知识图谱约束问答的强化学习推理技术*

毕 鑫[1], 聂豪杰[2], 赵相国[3], 袁 野[4], 王国仁[4]

[1](深部金属矿山安全开采教育部重点实验室(东北大学), 辽宁 沈阳 110819)

[2](东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

[3](东北大学 软件学院, 辽宁 沈阳 110169)

[4](北京理工大学 计算机科学与技术学院, 北京 100081)

通信作者: 聂豪杰, E-mail: qazxse2010@163.com

摘 要: 知识图谱问答任务通过问题分析与知识图谱推理, 将问题的精准答案返回给用户, 现已被广泛应用于智能搜索、个性化推荐等智慧信息服务中. 考虑到关系监督学习方法人工标注的高昂代价, 学者们开始采用强化学习等弱监督学习方法设计知识图谱问答模型. 然而, 面对带有约束的复杂问题, 现有方法面临两大挑战: (1) 多跳长路径推理导致奖励稀疏与延迟; (2) 难以处理约束问题推理路径分支. 针对上述挑战, 设计了融合约束信息的奖励函数, 能够解决弱监督学习面临的奖励稀疏与延迟问题; 设计了基于强化学习的约束路径推理模型 COPAR, 提出了基于注意力机制的动作选择策略与基于约束的实体选择策略, 能够依据问题约束信息选择关系及实体, 缩减推理搜索空间, 解决了推理路径分支问题. 此外, 提出了歧义约束处理策略, 有效解决了推理路径歧义问题. 采用知识图谱问答基准数据集对COPAR的性能进行了验证和对比. 实验结果表明: 与现有先进方法相比, 在多跳数据集上性能相对提升了 2%–7%, 在约束数据集上性能均优于对比模型, 准确率提升 7.8%以上.

关键词: 知识图谱; 约束路径推理; 约束问答; 强化学习

中图法分类号: TP18

中文引用格式: 毕鑫, 聂豪杰, 赵相国, 袁野, 王国仁. 面向知识图谱约束问答的强化学习推理技术. 软件学报, 2023, 34(10): 4565–4583. http://www.jos.org.cn/1000-9825/6889.htm

英文引用格式: Bi X, Nie HJ, Zhao XG, Yuan Y, Wang GR. Reinforcement Learning Inference Techniques for Knowledge Graph Constrained Question Answering. Ruan Jian Xue Bao/Journal of Software, 2023, 34(10): 4565–4583 (in Chinese). http://www.jos.org.cn/1000-9825/6889.htm

Reinforcement Learning Inference Techniques for Knowledge Graph Constrained Question Answering

BI Xin[1], NIE Hao-Jie[2], ZHAO Xiang-Guo[3], YUAN Ye[4], WANG Guo-Ren[4]

[1](Key Laboratory of Ministry of Education on Safe Mining of Deep Metal Mines (Northeastern University), Shenyang 110819, China)

[2](School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China)

[3](Software College, Northeastern University, Shenyang 110169, China)

[4](School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

Abstract: Knowledge graph based question answering (KGQA) analyzes natural language questions, performs reasoning over knowledge graphs, and ultimately returns accurate answers to them. It has been widely used in intelligent information services, such as modern search engines, and personalized recommendation. Considering the high cost of manual labeling of reasoning steps as supervision in the relation-supervised learning methods, scholars began to explore weak supervised learning methods, such as reinforcement learning, to

* 基金项目: 国家自然科学基金(62072087, 61932004, 62002054, 61972077, U2001211)

本文由“知识赋能的信息系统”专题特约编辑高宏教授、陈华钧教授、赵翔教授、李瑞轩教授推荐.

收稿时间: 2022-07-05; 修改时间: 2022-08-18, 2022-12-14; 采用时间: 2022-12-28; jos 在线出版时间: 2023-01-13

design knowledge graph based question answering models. Nevertheless, as for the complex questions with constraints, existing reinforcement learning-based KGQA methods face two major challenges: (1) multi-hop long path reasoning leads to sparsity and delay rewards; (2) existing methods cannot handle the case of reasoning path branches with constraint information. To address the above challenges in constrained question answering tasks, a reward shaping strategy with constraint information is designed to solve the sparsity and delay rewards. In addition, reinforcement learning based constrained path reasoning model named COPAR is proposed. COPAR consists of an action determination strategy based on attention mechanism and an entity determination strategy based on constraint information. Itis capable of selecting the correct relations and entities according to the question constraint information, reducing the search space of reasoning, and ultimately solving the reasoning path branching problem. Moreover, an ambiguity constraint processing strategy is proposed to effectively solve the ambiguity problem of reasoning path. The performance of COPAR is verified and compared using benchmark datasets of knowledge graph based question answering task. The experimental results indicate that, compared with the existing methods, the performance on datasets of multi-hop questions is relatively improved by 2%−7%; the performance on datasets of constrained questions is higher than the rival models, and the accuracy is improved by at least 7.8%.

Key words: knowledge graph; constrained path reasoning; constrained question answering; reinforcement learning

目前, 越来越多的学者将知识图谱[1,2]作为底层数据支撑应用在智能搜索问答[3]、智能对话[4]、个性化推荐[5]等领域. 其中, 问答是用户获取信息与知识的主流手段. 问答任务通过解析用户提出的问题, 向用户返回查询、推理结果. 传统的搜索引擎只能返回与查询关键字相关的内容排序列表, 用户需要逐条筛选自己需要的信息, 并寻找答案. 而知识图谱问答(knowledge graph based question answering, KGQA)[6,7]旨在返回问题的答案及其关联知识, 能够极大地提高用户的知识获取效率.

基于深度学习的 KGQA 技术大致分为关系监督[8]和弱监督[9,10]两类. 关系监督学习方法在每一跳推理过程中, 将与问题中的推理信息相似度评价最高的关系作为每一跳的预测结果. 然后, 这一跳的标记关系被用作监督信息来计算训练损失和更新网络参数. 当达到预定的停止关系时, 该方法会终止推理过程, 返回答案. 由于每一跳推理的关系已被人工标记, 这些方法在推理答案和推理路径方面都取得了很高的准确性. 但是, 对于每个训练样本, 推理路径上的所有关系都需要被人为标记. 对于大规模的知识图谱来说, 关系监督方法需要较为昂贵的代价准备训练数据集. 另一方面, 弱监督学习方法不需要人工标记关系, 其核心理念是, 根据行动空间的概率分布进行状态转移并搜索答案实体. 通过将问答推理任务建模为马尔可夫决策过程, 然后采用答案实体正确性作为监督信息, 并将终止奖励沿推理路径进行回溯式传播, 从而实现模型参数更新, 优化推理过程.

简单问答的研究集中在知识三元组序列匹配问题上, 目前已取得非常好的效果. 然而, 对于用户提出的一些比较复杂的问题, 问答模型往往需要以$\{(e_0,r_1,e_1),(e_1,r_2,e_2),\ldots,(e_{n-1},r_n,e_{answer})\}$为路径进行多跳推理才能找到答案, 弱监督方法只有在到达答案后才会获得最终奖励, 当推理路径过长时, 无法依靠奖励及时调整推理过程, 学习效率低. 同时, 简单问题的推理方法忽略了问题约束信息, 难以在路径分支推理中做出准确的选择, 严重影响了分支路径搜索的效率. 此外, 推理过程中还存在推理路径分支情况, 当出现同一约束信息作用在推理路径的不同中间节点时, 便会出现路径歧义分支. 因此, 针对简单问题的推理技术难以处理带有约束的复杂问题.

基于上述分析, 基于强化学习的知识图谱约束问答面临的严峻挑战可总结为两方面.

(1) 长路径推理导致稀疏且延迟的奖励. 基于强化学习的方法采用弱监督的方式进行训练, 只有当决策路径最后一步走到答案实体时, 才能得到正向奖励. 当知识图谱规模较大时, 搜索路径过长, 会产生奖励稀疏和延迟的问题;

(2) 难以处理约束问题推理路径分支. 对于带有约束的问题, 其推理路径在有约束的实体处必定存在分支, 而强化学习方法在路径推理时, 每个时间步只选择一个动作, 在路径分支情况下, 模型无法做出正确决策, 导致模型效率低下.

为了解决知识图谱约束问答的上述挑战, 本文提出一种基于强化学习的约束问答技术, 包括融合约束信息的奖励函数以及约束路径推理模型. 具体而言, 本文的贡献点主要包括以下 3 点.

(1) 对于奖励稀疏和延迟问题, 本文提出了融合约束信息的奖励塑形策略, 根据所选关系和实体约束信息与问题的相关性做出决策, 将推理过程中的每一步获取到的信息增益作为奖励塑形依据, 从而引导模型在做出正确决策的同时, 进一步提高模型训练效率;

(2) 对于带有实体或属性约束的问题, 本文提出了基于强化学习的约束路径推理模型(reinforcement learning based constrained path reasoning model, COPAR). 在关系路径推理时, 根据注意力机制计算关系的概率分布; 若选择的关系对应多个实体, 将各实体的约束信息作为实体选择的依据. 在选择完关系或约束之后, 对问题表示进行更新, 掩盖已选择的信息. 通过在当前步推理考虑约束信息, 缩短了推理路径. 另外, 设计了歧义约束处理策略, 提高了推理路径的准确性;

(3) 在多个数据集上进行实验, 与当前先进的基于强化学习的模型进行对比, 以预测准确率为指标, 对模型的性能进行验证. 另外, 本文还针对模型各部分设计了消融实验, 以证明不同模块对模型总体性能的影响.

本文第 1 节介绍知识图谱问答的相关方法和研究现状. 第 2 节将约束路径推理建模为马尔可夫决策过程. 第 3 节介绍本文提出的约束路径推理模型. 第 4 节通过对比实验验证了所提模型的有效性. 最后总结全文.

1 知识图谱问答相关工作

目前, 已有大量针对知识图谱问答的研究. 基于强化学习 RL (reinforcement learning)[11]的 KGQA 方法将推理路径的学习转化为序列决策问题, 与路径排序算法 PRA (path-ranking algorithm)相比缩小了搜索空间, 同时为推理提供可解释性. Xiong 等人提出的 DeepPath[12]最早将强化学习应用到知识图谱推理, 主要解决两个实体之间的关系预测和事实真假判断问题, 且该方法为每个关系训练其相应的模型, 可扩展性不强, 应用范围有限. Das 等人提出的 MINERVA[13]从问答角度对 DeepPath 进行的改进, 使用 REINFORCE[14]算法实现了端到端的知识图谱简单问答. 但该方法将问题结构化为⟨*head*,*relation*,?⟩的形式, 限制了其处理复杂问题的能力.

DeepPath 和 MINERVA 采用的都是 Policy Gradient 方法, 在基于价值的方法这一分支下, Shen 等人提出了 M-Walk[15]模型, 该模型使用 Q-learning, 以 off-policy 方式改进强化学习模型; Wan 等人提出了 HRL[16]模型, 是对 Policy Gradient 的一种改进, 整个推理过程被分解为两级的强化学习策略层次结构, 用于对历史信息进行编码并学习结构化的动作空间, 旨在解决多重语义的问题; Wang 等人提出了 ADRL[17]框架, 它通过对深度学习和关系推理的结构化理解来提高传统方法的效率, 使用 Actor-Critic 算法对整个框架进行优化; Zhang 等人[29]提出了一种自适应强化学习(ARL)框架来解决复杂问答, 通过设计自适应路径生成器生成关系路径, 以指导智能体到达目标实体.

在搜索空间优化方面, Qiu 等人针对动作搜索空间过大的问题, 提出了 SRN (stepwise reasoning network)[18]模型, 通过注意力机制获取到当前步可能关注的关系, 通过计算动作空间与问题中关系的相关度, 采用波束搜索的方法来缩减动作空间. 另外, 考虑到模型偶然到达目标节点的情况, 即模型找到最终答案, 但采取的路径与问题当中的关系并不相关, 这会给强化学习模型的学习带来一定的干扰, Lin 等人[19]提出了 Action Dropout 方法, 从一定程度上缓解了这一问题. Kaiser 等人[28]基于 Actor-Critic 提出了 CONQUER, 针对知识图谱上的对话式问答任务, 引导多个智能体在知识图谱上进行并行推理, 并以问题重述的方式构建隐式反馈奖励.

在奖励塑形方面, MINERVA 的奖励函数设置虽然从一定程度上反映了学习目标, 但该函数会带来奖励稀疏的问题, 给强化学习模型的学习带来一定难度, 很多使用强化学习方法解决问答问题的工作中都在探索奖励函数该如何设定. Lin 等人[19]采用奖励塑形的方法, 通过预训练的单跳嵌入模型来估计路径的奖励值, 以减少负面监督的影响; SRN[18]在此基础上提出了一种 Potential-based[20]奖励塑形方法, 将历史路径对于问题中信息的覆盖度作为衡量问题解决程度的方法, 从而无需借助其他方法来缓解弱监督带来的奖励稀疏问题.

基于强化学习的方法往往采用弱监督的形式进行训练, 不需要中间推理步骤的人工标注数据. 但当面对复杂的约束问题时, 现有的奖励塑形方法不能高效处理含约束信息的路径分支. Lin 等人[19]只考虑到当前实体

后一跳推理所带来的奖励, 忽略了当前跳推理实体的约束信息, 在面对无关分支或歧义分支时, 无法做出准确的选择. SRN[18]采用问题信息的历史路径覆盖度作为衡量标准, 在遇到约束分支路径的覆盖度高于正确路径时, 容易将覆盖度更高的错误路径作为当前的推理关系, 从而产生错误的推理结果. 本文所设计的奖励塑形函数能够同时考虑历史推理路径和约束信息对问题的覆盖度, 以保证模型推理结果与问题中的未推理关系具备高度语义相关性, 并且避免出现约束路径推理错误.

2 约束问答推理任务建模

2.1 问题定义

知识图谱问答任务根据用户发起的自然语言问题, 首先分析出问句中的实体或属性, 将识别出来的实体链接至知识图谱中相对应的节点, 从识别出的实体中确定主题实体, 并在与其相关的知识图中进行路径推理得到最终答案. 本文假设所要寻找的答案在知识图谱中且实体识别与实体链接部分已通过相应技术完成, 主要研究搜索答案的路径推理过程.

相比知识图谱简单问答, 知识图谱约束问答包含约束信息, 导致其相关概念与推理过程具有特殊性. 因此, 首先给出以下知识图谱约束问答相关定义.

定义 1(约束路径推理). 约束路径推理是指知识图谱约束问答任务的推理过程. 将问题中识别出的实体、属性值区分为主题实体 e_{te}、路径实体集 E_p、约束实体集 E_c. 问题约束中对应知识图谱实体的约束称为实体约束; 问题约束中对应知识图谱中主路径实体属性值的约束称为属性约束. 从问题中的主题实体 e_{te} 所对应的节点出发, 在知识图谱 G 中根据问题 q 的信息以及约束实体集 E_c 进行推理, 返回由长度为 L 的决策路径 h 以及节点 c^* 约束构成的约束推理路径.

定义 2(知识图谱约束问答). 针对给定包含约束信息的自然语言问题 q, 知识图谱问答旨在知识图谱中进行约束路径推理, 并依据其返回的推理路径获取最终的尾实体作为答案返回给用户.

而对于带有约束的问题, 通常不能通过一个三元组事实推理得到答案, 且多个三元组事实通常也不是简单的链式关系, 在问答推理过程中往往包含分支, 现有针对简单多跳问题的问答模型不具有处理复杂约束问题的高效性.

图 2.1 展示了一个带约束的自然语言问题“Which film starred by Forest Whitaker is directed by Mark Rydell?”及回答该问题所需要的知识子图实例. 该实例中, 从主题实体节点 Mark Rydell 出发, 通过关系 directed_film 选择尾实体集合[Even Money,TheFox]中的一个节点作为当前步的推理结果. 现有的简单问答方法大多从尾实体集合中随机选择一个, 然后继续进行下一步探索. 当训练次数足够多时, 强化学习有能力学习到正确的推理路径, 但这需要大量的计算. 即便刚好选中答案实体 Even Money, 在基于弱监督的情况下, 智能体得到了较高的奖励, 但此时答案路径中并不包含“starred by Forest Whitaker”这一约束信息, 推理路径的语义不完整, 训练得到的结果也随之受到影响. 若要保证智能体按路径⟨Mark Rydell,directed_film,Even Money⟩探索到正确的答案实体, 推理过程需要满足实体约束需求, 即针对 Even Money 节点匹配知识三元组⟨Even Money,acted_film,Forest Whitaker⟩.

当问题变为“When was the film starred by Forest Whitaker and directed by Mark Rydell released?”时, 假设主题实体仍为 Mark Rydell, 要得到正确答案, 则需按需按路径⟨Mark Rydell,directed_film,Even Money⟩, ⟨Even Money,released_time,2006-05-08⟩进行推理. 其中, 节点 Even Money 带有实体约束⟨Even Money,acted_film, Forest Whitaker⟩. 然而, 现有基于强化学习的问答模型在探索此类路径的过程中, 假设可以到达实体 Even Money, 此时下一跳推理路径有两个可选项, 即⟨Even Money,acted_film,Forest Whitaker⟩和⟨Even Money, released_time,2006-05-08⟩. 若选择关系为 released_time 的路径, 则直接到达答案节点, 此时所包含的语义内容与问题相比不够全面; 若选择关系为 acted_film 的路径, 则还需返回至 Even Money 节点, 再选择下一个关系 released_time, 加长了推理路径.

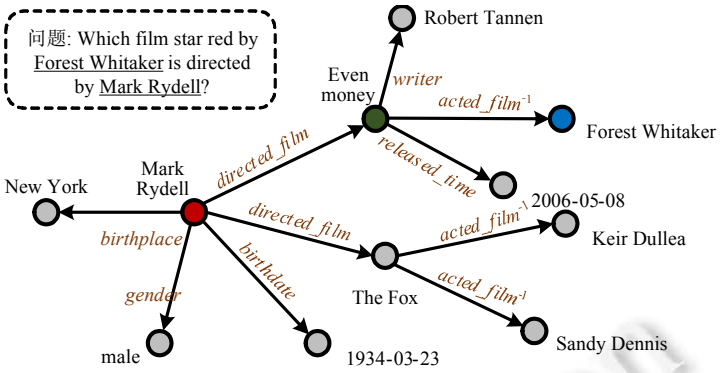


图 2.1 一个用于回答带约束问题的知识图谱子图示例

从上述实例分析可知: 现有知识图谱问答推理模型在面临一对多关系以及推理路径产生分支时, 处理效率显著下降. 若问题中包含该关系对应的多个实体的约束信息, 现有模型在推理过程中无法预先得知该约束, 因而在一个关系对应多个实体时, 无法依据约束信息准确选取实体. 当推理路径中某个实体具有约束时, 推理路径产生分支, 模型面临两种选择: 第 1 种可以先去处理约束, 返回后继续寻找答案, 拉长推理路径; 另一种方法直接选择下一跳关系进行推理, 但该方法到达答案节点后, 推理路径所包含的语义信息不完整, 可解释性差.

2.2 马尔可夫决策过程建模

本研究将知识图谱问答的路径推理视为序列决策问题, 在路径推理过程中, 当前时刻 t 之后所处的状态与时刻 t 之前的状态无关, 具有马尔可夫性, 可将其建模为马尔可夫决策过程. 相对应地, 智能体就是本文要学习的网络模型, 该网络模型用来对动作空间中的动作进行评估, 可得到每个动作的选取概率, 环境则包括除决策网络以外的相关因素, 即知识图谱 G 和自然语言问题 q.

得知了环境和智能体这一组交互对象, 对于马尔可夫决策过程的 4 个主要元素: 状态 S、动作空间 A、状态转移概率 P 及奖励 R, 在知识图谱问答任务中, 各元素的表示如下.

(1) 状态空间 S

状态空间 S 指的是环境中所有状态 s 的集合. 在时间步 t, 状态 s_t 用来表示智能体在时间步 t 的观测值, 可根据观测值进行动作概率的计算. 在约束问答任务中, 当前状态可表示为 $s_t=(q,e_{te},e_t,h_t,c,\boldsymbol{q}_t)\in S$, 其中, e_{te} 为给定自然语言问题 q 中的主题实体, e_t 表示在路径推理过程中智能体在第 t 步到达的实体, h_t 表示智能体在时间步 t 之前时刻的历史推理路径, c 中记录了推理过程中每个时间步所到达的实体对应的约束信息, $\boldsymbol{q}_t$ 为针对上一步选择的动作及所选择实体的约束对问题更新后的问题表示. 问题 q 与主题实体 e_{te} 可认为是全局信息, 推理过程中不发生改变. 本研究中, 定义初始状态 $s_0=(q,e_{te},e_t,h_0,c,q_0)$, 其中, $h_0=\varnothing$; $c=[\cdot]*L$, L 为推理路径长度; $\boldsymbol{q}_0=QuestionEncoder(q)$为问题 q 的初始编码.

(2) 动作空间 A

动作空间 A 指的是所有动作 a 的集合. 在状态 s_t 可采取的所有动作的集合为当前的动作空间 $A(s_t)$. 在传统知识图谱问答任务中, 将动作空间表示为节点 e_t 连接的所有出边, 包括关系及对应的尾实体, 即 $A(s_t)=\{(r,e)|(e_t,r,e)\in G\}$. 另外, 由于在实际任务中, 问题的推理路径长度是不固定的, 因而为每个实体引入“self_loop”关系连接实体本身, 作为当前可行动作空间中的一个动作, 当路径推理已到达答案节点但还未达到推理路径长度 L 时, 可选择“self_loop”自环作为路径推理的终止条件.

当知识图谱规模较大或同一关系对应多个实体时, 这种动作空间设计需要大量的计算和存储空间. 在面对一对多关系对应的实体集 $E'=\{e|(e_t,r_t,e)\in G\}$中的实体选择时, 需从中选择带约束的实体, 并不能根据动作 $a'=(r',e')$中的实体信息 e'对约束进行判断. 因而, 本文将智能体当前所在节点连接的所有出边的无重复关系集合表示为当前的动作空间, 即 $A(s_t)=\{r|(e_t,r)\in G\}$, 其中, $r\in R$ 表示与实体 e_t 的连接边对应的关系. 该方法有效地

将每个实体的动作空间都控制在一定范围内, 同时可对一对多关系对应的实体进行区分, 进而选择满足约束的实体.

(3) 状态转移概率 P

根据传统方法状态与动作的定义, 若智能体在 t 时刻采取动作 $a_t=(r^*,e^*)$, 则可以确定, 状态一定可以转换至 $s_{t+1}=(q,e_{te},e^*,h_{t+1},c,q_{t+1})$, 其中, $h_{t+1}=h_t\cup a_t$. 也就是说, $P(s_{t+1}|s_t,a_t)=1$. 针对本文采用的动作空间设计方法, 选择完动作之后只能确定当前跳的关系, 若关系对应多个实体, 则实体的选择不能确定. 本文针对约束情况, 根据自然语言问题 q 及获取到的可能的约束对状态转移概率 P 进行计算.

(4) 奖励函数 R

一般来讲, 奖励函数是为提高模型效率人为设定的. 在传统方法中, 只有当智能体最终到达目标节点 e_{ans} 时, 才能得到一个正值奖励, 在推理过程中或推理最终未到达目标节点, 则不会得到奖励, 如公式(2.1)所示:

$$R(s_t)=\begin{cases}1, & \text{if } e_t=e_{ans}\\ 0, & \text{otherwise}\end{cases} \tag{2.1}$$

然而, 该奖励函数只有在到达答案节点后才能得到正值奖励. 如果当前状态的动作空间较大, 则在训练前期, 通过随机访问的方式搜索到主推理路径节点的概率极小. 因此, 模型推理效率低下, 并且很有可能导致模型无法收敛. 为了解决这个问题, 本文提出奖励塑形策略, 设计了融合约束信息的奖励函数.

2.3 融合约束信息的奖励函数

本文采用基于潜力的奖励塑形(potential-based reward shaping)[20]技术为智能体的每一步动作提供奖励. 基于潜力的奖励塑形是指: 若一个奖励塑形函数 $F:S\times A\times S\to R$ 是基于潜力的, 需满足: 对交互过程中所有决策 $\langle s\neq s_T,a,s'\rangle$, 存在势能函数 $\phi:S\to R$, 使得 $F(s,a,s')=\gamma\phi(s')-\phi(s)$成立. 其中, s_T 代表终止状态, s'表示状态 s 之后的下一个状态, $\phi:S\to R$ 表示势能函数ϕ为状态 S 到奖励 R 的映射, γ为折扣因子. 势能函数$\phi(S)$一定程度上能够反映当前状态与目标状态之间的距离, 若智能体选择了正确的动作或与正确关系语义相近的动作, 其势能才会增加. 根据以上定义, 对于每一个状态转移过程$\langle s,a,s'\rangle$, 都会有一个额外的塑形奖励$\gamma\phi(s')-\phi(s)$, 使得交互过程的奖励值变得稠密, 加速模型收敛的同时, 可保证使用塑形奖励与仅使用主线奖励智能体的目标和学习到的最优策略是一致的.

基于本文动作空间的设计方法, 单步奖励可表示为从当前状态 s_t 选择动作 a_t、实体 e_{t+1}、约束 $c[t]$转移至状态 s_{t+1} 得到的回报, 形式化表示为 $R(s_{t+1}|s_t,a_t)$. 在问答任务中, 智能体采取一个正确的动作对应的关系应与问题 q 中的某个关系在语义上存在高度相关性, 状态转移时获取到的约束信息 c 也应与问题 q 中的约束关系及实体相对应. 因此, 本文将潜力函数用智能体已探索的历史决策路径 h_t 和约束信息 c 对自然语言问题 q 的向量化表示 $\boldsymbol{q}$ 的语义涵盖度来表示, 计算公式如公式(2.2)所示:

$$\phi(s_t)=\begin{cases}\sigma(\cos(\boldsymbol{\tau}_t,\boldsymbol{q})), & t>0\\ 0, & t=0\end{cases} \tag{2.2}$$

其中, $\boldsymbol{\tau}_t$表示将约束信息c融入历史决策h_t的约束路径$\boldsymbol{\tau}_t$通过问题编码方法编码后的向量表示, $\sigma(\cdot)$为激活函数.

根据基于潜力的塑形奖励函数的定义及本文任务中对势能的定义, 可以得到奖励塑形函数 $F_\phi(s_t,a_t,s_{t+1})=\gamma\phi(s_{t+1})-\phi(s_t)$. 根据 Ng 等人[20]的推理可知, 将奖励塑形函数与主线奖励函数相结合, 智能体的整体策略目标保持不变. 因此, 本文奖励函数如公式(2.3)所示:

$$R^*(s_t,a_t,s_{t+1})=R+F_\phi(s_t,a_t,s_{t+1}) \tag{2.3}$$

其中, R 为公式(2.1)所示的主线奖励.

3 约束路径推理模型

本文设计了针对实体或属性约束问题的路径推理模型 COPAR. 与传统的基于强化学习的问答模型相比, COPAR 除了基本的关系路径决策外, 还包含了对约束的处理策略, 能够对约束问题中的主题实体及约束实体集进行区分, 从而依据问题约束信息缩减推理搜索空间. 在动作选择策略中, 基于注意力机制设计了融合约

束信息的实体选择概率计算方法, 通过约束信息以及问题对各个候选关系的关注度选取当前推理关系. 另外, 在选择完动作或约束之后, COPAR 能够依据推理关系及约束关系对问题的嵌入表示进行实时更新, 从而在选择下一跳关系时, 更加关注未推理的语义信息. COPAR 的整体框架如图 3.1 所示.

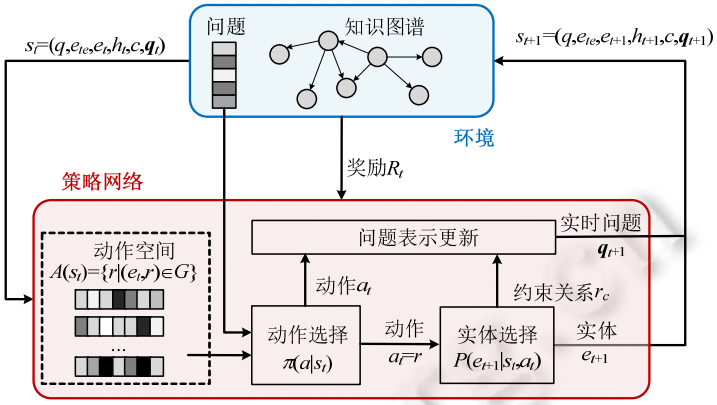


图 3.1 约束路径推理模型框架

图 3.1 展示了对问句进行实体识别、实体链接、主题实体及约束实体集区分之后, 环境与智能体的互动过程. 首先观察到当前状态 s_t=(q,e_{te},e_t,h_t,c,$\boldsymbol{q}_t$), 从当前实体 e_t 处得到当前可行的动作空间 $A(s_t)$. 策略网络对当前问题表示 $\boldsymbol{q}_t$、动作对应的关系 $\boldsymbol{r}$、约束实体集 E_c 进行分析计算, 确定当前跳的关系、实体及约束, 对问题进行更新后, 转移至下一个状态, 同时, 环境给出奖励值 R_t 作为反馈, 随后进行下一跳推理.

3.1 主题实体与约束实体集区分

现有方法只简单地选取问题中识别出来的实体或属性值中的第 1 个实体作为主题实体, 对于相对而言比较复杂的约束问题, 这种方法往往致使整个推理路径变长或推理过程变复杂. 如问题“Who plays defender was born in Reading whose WOEID is 32997?”, 使用开源的 Berkeley Neural Parser[21]库对其进行解析. 由标签树的分析可知, 实体 Reading 位于该问题的主干. 对于已识别出的该问题中的实体及属性值 E_s=[Reading,defender, 32997], 此时不论选取 defender 还是属性值 32 997, 都将导致推理路径变长, 处于问句主干的 Reading 无疑是主题实体的最佳选择.

对可以识别出主干实体的问题, 本文根据开源的 Berkeley 神经解析器得到问题的标签树, 并将从标签树中识别出的动词短语(标签‘VP’)或介词短语(标签‘PP’)中第 1 个名词作为主干实体, 本文将名词定义为标签‘NP’且子标签均为‘NNP’的词. 若问句中包含主干实体, 则选取主干实体为主题实体, 若没有, 则选取第 1 个实体为主题实体.

在确认主题实体及约束实体集后, 将问题中文本进行对应的符号替换, 建立问题中词语的语料库, 进而对其进行编码. 考虑到基于 RNN 的方法只能获取前后两个单词的信息, 而对于比较复杂的约束问题, 约束信息与实体的间隔可能非常大, 只考虑前后两个词的语义信息远远不够. Transformer 编码器通过自注意力机制考虑句子中每个词对当前词的影响, 因而本文工作采用 Transformer 的编码器部分对问题及知识图谱中关系进行编码.

3.2 策略网络模型

策略网络中, 主要包括动作选择、实体选择及实时问题更新过程. 从环境中可以观察到状态 s 及动作空间 $A(s_t)$, 对问题文本 q 进行预处理后, 得到主题实体 e_{te} 及约束实体集 E_c, 通过 Transformer 的编码器模块将问题及关系分别表示为低维空间中的向量; 然后根据各要素的向量表示, 在动作选择部分, 经过一系列计算, 得到当前状态下动作的概率分布 $\pi(a|s_t)$, 据此分布采样一个动作 a_t, 并针对动作对应的关系 r_t 对问题 $\boldsymbol{q}_t$ 进行实时更新; 在实体选择部分, 通过约束实体集 E_c 获取可能的约束信息, 并计算当前的实体选择概率 $P(e_{t+1}|s_t,a_t)$, 智

能体根据这一概率分布, 选择一个实体 e_{t+1}, 针对获取的约束关系 r_c 对问题 $\boldsymbol{q}_t$ 进行实时更新得到 $\boldsymbol{q}_{t+1}$, 进而转移至下一个状态 $s_{t+1}=(q,e_{te},e_{t+1},h_{t+1},c,\boldsymbol{q}_{t+1})$, 得到相应的奖励 R_t, 并进行下一跳探索. 本节将介绍动作选择、实体选择及问题更新的具体实现.

3.2.1 基于注意力机制的动作选择策略

在得到给定的自然语言问题 q、当前动作空间 A、历史路径 h_t 的向量化表示 $\boldsymbol{h}_t$ 及当前状态 s_t 后, 智能体需要经过对动作空间 A 中动作的分析, 我们利用注意力机制计算出问题对当前动作空间中所有动作的关注度, 选出最优的动作 a_t. 本文首先将当前问题表示 $\boldsymbol{q}_t$ 与动作空间 $A(s_t)$中的动作 a 通过注意力机制进行交互, 获取当前状态下问题 $\boldsymbol{q}_t$ 中每个词语针对动作 a 的关注度, 由此得到具有注意力权重的问题表示 $\boldsymbol{q}'_t$, 根据问题 $\boldsymbol{q}'_t$ 计算动作 a 的语义分数 $Score(a,q)$, 根据该方法得到动作空间中所有动作 $a=r\in A(s_t)$的语义分数之后, 对其归一化得到动作的概率分布 $P(a|s_t)$. 其网络结构如图 3.2 所示.

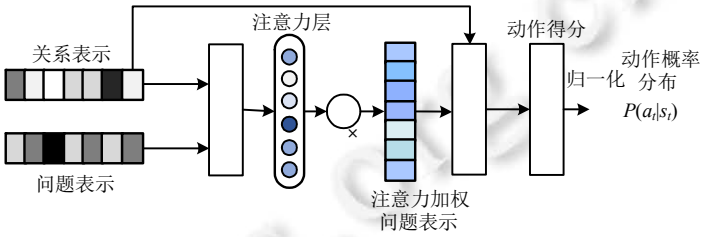


图 3.2 动作选择策略网络

在当前状态 $s_t=(q,e_{te},e_t,h_t,c,\boldsymbol{q}_t)$下, 智能体通过在环境中观察获取到当前的动作空间 $A(s_t)=\{r|(e_t,r)\in G\}$, 对于所有的动作 $a=r\in A(s_t)$, 计算其对应的关系 r 与当前问题 $\boldsymbol{q}_t=(\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_n)$中的每个单词向量的相似度分数, 将得到的结果通过归一化操作, 便可得到当前关系对应的问题中每个单词的注意力权重$\boldsymbol{\Lambda}=(\alpha_1,\alpha_2,\ldots,\alpha_n)$, 由此得到对关系 r 感知的问题表示 $\boldsymbol{q}'_t=(\alpha_1\boldsymbol{w}_1,\alpha_2\boldsymbol{w}_2,\ldots,\alpha_n\boldsymbol{w}_n)$. 在此基础上, 计算动作 a 的语义分数 $Score(a,q)$, 其计算公式如(3.1)所示:

$$Score(a,q)=\boldsymbol{r}\boldsymbol{W}_2\boldsymbol{q}_t'^{\mathrm{T}} \tag{3.1}$$

其中, $\boldsymbol{r}$ 表示当前动作空间 $A(s_t)$中动作 a 对应的关系 r 的向量表示, $\boldsymbol{W}_2$ 表示网络参数. 关系 r 感知的问题表示 $\boldsymbol{q}'_t$ 计算公式如下:

$$\boldsymbol{q}'_t=\Lambda^{\mathrm{T}}\otimes\boldsymbol{q}_t \tag{3.2}$$

其中, ⊗表示两向量对应位置相乘. Λ表示计算得到的关系 r 对问题 $\boldsymbol{q}_t$ 中每个单词的注意力权重, 计算公式如下:

$$\Lambda=SoftMax(\boldsymbol{W}_1\cdot\cos(\boldsymbol{r}\times\boldsymbol{q}_t^{\mathrm{T}})) \tag{3.3}$$

其中, cos(·)表示向量矩阵相乘计算向量之间相似度, $\boldsymbol{W}_1$ 表示网络参数.

通过计算语义分数, 可得到关系 r 与当前问题 $\boldsymbol{q}_t$ 的匹配程度. 获取到所有关系的语义分数之后, 将其通过 *SoftMax* 层进行归一化, 便可得到动作空间 $A(s_t)$中每个动作 a_t 的概率分布 $\pi(a_t|s_t)$, 如公式(3.4)所示:

$$\pi(a_t\mid s_t)=\frac{\exp(Score(a_t,q_t))}{\sum_{a\in A(s_t)}\exp(Score(a,q_t))} \tag{3.4}$$

由此, 可根据以上得到的动作概率分布 $\pi(a_t|s_t)$, 从动作空间 $A(s_t)$中采样一个动作 a_t, 确定当前跳选择的关系 $r_t=a_t$.

3.2.2 基于约束的实体选择策略

当根据策略 π选择的关系 $r_t=a_t$ 对应多个实体 $E'=\{e'_1,e'_2,\ldots,e'_m\}$ 时, 需考虑实体的选择. 对于简单问答任务而言, E'中多个实体可以无差别选择, 多个实体选择概率相同并不影响推理的准确性. 但对于自然语言问题 q 在实体集 E'中的某个实体有约束的情况, 一旦实体没有选择正确, 便会级联到后续推理, 即便关系路径正确, 得到的答案也非正确答案, 极大地降低了模型的准确度, 且容易导致正例过少, 模型不易收敛. 因而, 在选择

完关系后, 若对应多个实体, 则需根据实际处理的问题分析到达多个实体的可能性作为实体选择的依据.

本文提出依据一对多关系对应的实体集 E'中实体满足约束的情况来计算实体选择概率的方法. 本文将约束实体集 E_c 是否为空作为问题是否约束问题的条件: E_c 为空, 则表示该问题非约束问题; E_c 不为空, 则在每一跳关系选择完之后, 寻找关系对应的实体集 E'中的实体可能的约束, 并对每个实体对应的约束信息计算得分.

具体而言, 约束寻找过程首先筛选出候选实体集 E'与约束实体集 E_c 之间的所有连接边作为约束集 *constraint*, 即 $constraint=\{(r_c,e_c)|(e',r_c,e_c)\in G,e'\in E',e_c\in E_c\}$, 并以候选实体集 E'中实体 e'为单位, 采用公式(3.1)–公式(3.3)计算约束关系 r_c 与当前问题 q 的语义分数作为约束得分, 得到当前约束集 *constraint* 中实体 e'对应的每个约束的得分 $Score_c$, 将该实体对应的所有分数较高(大于阈值ε)的约束得分累加作为该实体的最终得分.

本文对实体集 E'中每个实体 e'的约束进行语义分数计算后叠加至实体 e', 通过该方法得到每个实体的最终得分 $Score=[Score_{c1},Score_{c2}]$. 在进行约束分数叠加时, 考虑到约束获取过程中可能混入与问题不相关的约束关系, 如实体 Hello Dolly!的约束(lyrics_by,Louis Armstrong). 本文将候选实体集中同一实体上分数较高的约束分数进行叠加, 当约束较多时, 可有效地过滤掉约束集中一部分与问题无关的约束. 最后通过对最终得分 *Score* 进行归一化操作, 便可得到实体集 E'中每个实体的选择概率 $P(e_{t+1}|s_t,a_t)=SoftMax(Score)$及实体对应的约束. 算法 3.1 展示了基于约束的实体选择概率的计算过程.

根据以上过程得到的实体选择概率分布, 本文选取 E'中概率最大的实体 e'作为当前跳选择的实体, 即 $e_{t+1}=e'$. 根据 e'在实体集中的相对位置, 可在约束集 *constraint* 中获取到对应位置的 e'的约束. 上述过程中, 在得到候选实体集 E'中每个实体的选择概率 P 的同时, 过滤掉约束集 *constraint* 中与问题不相关的部分约束, 但还不能保证所有约束均与问题相关. 对于图 3.4 中的例子, 假设模型可根据实体选择概率 $P(e_{t+1}|s_t,a_t)$选择实体 e_{t+1}=Hello,Dolly!, 并获取到对应的约束集 $c_t'=[(\text{performed\_film}^{-1},\text{Louis Armstrong}),(\text{lyrics\_by},\text{Louis Armstrong})]$, 此时, 约束实体 Louis Armstrong 通过 performer, lyrics_by 多个关系作用于实体 e_{t+1}, 具有歧义性.

针对上述实体上的歧义约束问题, 在确定当前跳选择的实体 $e_{t+1}=e'$并获取到其对应的约束集 c_t 后, 判断约束集是否包含实体上的歧义约束. 即: 若约束集中仍包含多条约束实体 e_c 相同的约束, 则存在实体上的歧义约束, 本文选择约束实体为 e_c 的所有约束中得分最高的约束关系 r_c 作为实体 e_{t+1} 上约束实体为 e_c 的唯一约束(r_c,e_c).

由此, 在选择完关系 r_t 之后, 便可计算对应实体集 E'的选择概率分布 P, 根据 P 选择相应实体 e_{t+1} 后, 并针对 e_{t+1} 对应的约束 c_t' 存在实体上歧义约束的情况对歧义约束进行处理, 最终得到无实体上歧义的约束集 c_t.

3.2.3 实时问题更新

复杂问题在每一跳推理时都有不同的注意力分布. 在每一跳选择完关系、实体及其对应的约束之后, 应针对路径关系及约束关系对问题进行更新, 从而弱化问题中已推理的关系信息, 避免在下一跳决策时受到已推理信息的干扰, 从而提高关系选择的准确率. 具体而言, 本文首先计算已选择的关系 r_t或约束关系 r_c对问题 $\boldsymbol{q}_t=(\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_n)$中每个单词的注意力权重$\Lambda=(\alpha_1,\alpha_2,\ldots,\alpha_n)$, 将注意力权重值较高的位置提取出来, 并重新计算与关系高度相关的词的注意力权重 α_i', 其公式如(3.5)所示:

$$\alpha_i'=\begin{cases}\dfrac{\alpha_i}{\sum\alpha_i}, & \alpha_i\in\left\{\alpha_i\,|\,\alpha_i>\dfrac{1}{n}\right\}\\ 0, & \text{otherwise}\end{cases},\ i\in[1,n] \tag{3.5}$$

其中, n 表示问题长度. 由此得到问题 q_t 中与关系 r_t 相关的词语的注意力权重分布 $\Lambda'=(\alpha_1',\alpha_2',\ldots,\alpha_n')$.

本文将由多个单词组成的关系表示为多个词向量之和, 即 $\boldsymbol{r}=+\boldsymbol{w}_{r1}+\boldsymbol{w}_{r2}+\boldsymbol{w}_{rm}$, 其中, $\boldsymbol{w}_r$ 表示关系 $\boldsymbol{r}$ 中对应位置的单词向量, m 表示关系长度. 对应于增加了注意力权重Λ'的问题 $\boldsymbol{q}_t'=(\alpha_1'\boldsymbol{w}_1,\alpha_2'\boldsymbol{w}_2,\ldots,\alpha_n'\boldsymbol{w}_n)$, 此时可将 $\boldsymbol{q}_t'$理解为只包含关系 r_t 相关的带权重的词信息, 由此可根据公式(3.6)得到关系 r_t 相关的每个词的向量表示:

$$\boldsymbol{w}_i'=\begin{cases}\dfrac{\boldsymbol{r}_t-\sum_{k=1,k\neq i}^{n}\alpha_k'\boldsymbol{w}_k}{\alpha_i'}, & \alpha_i\neq 0\\ [0]*d, & \text{otherwise}\end{cases} \tag{3.6}$$

其中, $\boldsymbol{r}_t$ 为关系 r_t 的向量表示, d 表示词向量维度, $\boldsymbol{w}_k$ 表示当前问题 q_t 中第 k 个单词的向量表示, n 表示问题长度. 由此得到问题 q_t 中与关系 r_t 相关的词语的向量表示 $\boldsymbol{q}^{\Delta}=(\boldsymbol{w}'_1,\boldsymbol{w}'_2,...,\boldsymbol{w}'_n)$.

为了屏蔽问题中已完成推理的关系 r_t, 将当前跳问题 $\boldsymbol{q}_t$ 中减去 $\boldsymbol{q}^{\Delta}$, 如公式(3.7)所示, 以作为下一次问题更新计算时的输入:

$$\boldsymbol{q}_t^*=\boldsymbol{q}_t^*-\boldsymbol{q}^{\Delta} \tag{3.7}$$

本文通过上述方法, 在每一跳选择完推理关系 r_t 及确定实体上的约束集 $c_t=\{(r_{c1},e_{c1}),\ldots,(r_{cl},e_{cl})\}$($l$ 为约束个数)之后, 分别根据所选择的关系 $r_t,r_{c1},\ldots,r_{cl}$ 及最后一次更新的问题表示 q_t^* 进行更新, 可在下一次关系选择之前将已选择的关系信息掩盖, 从而提高关系选择准确度.

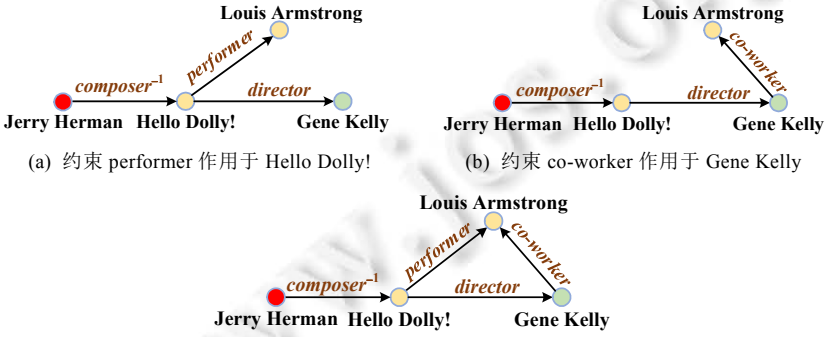
在通过策略网络中动作选择部分确定当前跳动作 a_t 后, 实体选择部分确定下一跳实体 e_{t+1} 及其对应的约束 c_t, 问题更新部分针对动作对应的关系 r_t 及约束关系 r_c 对问题进行更新得到 $\boldsymbol{q}_{t+1}$, 状态便转移至 $s_{t+1}=(q,e_{te},e_{t+1},h_{t+1},c,\boldsymbol{q}_{t+1})$.

3.3 路径歧义约束处理策略

在得到推理路径 τ 及约束集 c 之后, 仅采用实体上的歧义约束处理方法无法解决路径歧义约束, 即同一约束实体作用在推理路径中多个实体的问题. 例如: 对于约束问题“Who directed the film which was composed by Jerry Herman and performed by Louis Armstrong?”, 假设模型最终生成路径⟨Jerry Herman,composed_film,Hello Dolly!⟩, ⟨Hello Dolly!,directed_film,Gene Kelly⟩, 每个实体上对应的约束分别为[·], [(performed_film^{-1},Louis Armstrong)], [(co-worker,Louis Armstrong)]. 可以看出: 约束实体 Louis Armstrong 分别作用于推理路径的两个实体 Hello Dolly!及 Gene Kelly 上, 且实体 Gene Kelly 上的约束(co-worker,Louis Armstrong)与问题不相关.

路径歧义约束在推理路径生成过程中无法避免, 在得到推理路径 $\tau=s_1,a_1,s_2,a_2,\ldots,s_L,a_L$ 及各实体上的约束情况 c 后, 对存在歧义的约束进行处理. 具体而言, 首先对约束集 c 中的每个约束(r_c,e_c)进行统计, 若存在同一约束实体作用于路径中不同实体上的情况, 则认为该约束具有歧义性, 记录该约束实体出现的位置, 并一一列举具有歧义的约束及对应位置的可能的约束, 通过计算历史路径及约束情况组成的带约束的推理路径 $\boldsymbol{\tau}_c$ 与问题 $\boldsymbol{q}$ 的相似度, 得分最高的约束情况作为该路径的最终约束.

对于路径⟨Jerry Herman,composed_film,Hello Dolly!⟩, ⟨Hello Dolly!,directed_film,Gene Kelly⟩及各实体对应的约束[·], [(performed_film^{-1},Louis Armstrong)], [(co-worker,Louis Armstrong)], 可判断出约束实体 Louis Armstrong 同时作用于该路径的不同实体 Hello Dolly!及 Gene Kelly, 存在路径约束歧义. 因而, 首先记录推理路径中的实体相对位置作为约束存在的位置, 然后根据约束实体 Louis Armstrong 在整个推理路径中至少出现一次的规则, 列举约束实体在路径中可能出现的情况, 如图 3.3 所示.



(a) 约束 performer 作用于 Hello Dolly!　(b) 约束 co-worker 作用于 Gene Kelly

(c) 两种约束同时考虑

图 3.3　路径歧义约束情况实例

根据图 3.3 所示的 3 种可能情况, 可以得到 3 条带约束的推理路径, 见表 3.1.

表 3.1 3 种情况对应的约束路径

可能情况	约束路径 τ_c
图 3.3(a)	Jerry Herman, composed_film, Hello Dolly!, performed_film^{-1}, Louis Armstrong, directed_film^{-1}, Gene Kelly
图 3.3(b)	Jerry Herman, composed_film, Hello Dolly!, directed_film^{-1}, Gene Kelly, co-worker, Louis Armstrong
图 3.3(c)	Jerry Herman, composed_film, Hello Dolly!, performed_film^{-1}, Louis Armstrong, directed_film^{-1}, Gene Kelly, co-worker, Louis Armstrong

在得到每种约束情况相应的约束路径τ_c之后, 采用问题编码方法对τ_c进行编码, 然后根据公式(3.8)计算两者的相似度:

$$sim(\tau_c,q)=\cos(\boldsymbol{\tau}_c,\boldsymbol{q}) \tag{3.8}$$

其中, $\boldsymbol{\tau}_c$、$\boldsymbol{q}$分别为约束路径τ_c及问题q的向量表示. 最终, 采取相似度较高的约束路径作为最终的推理路径.

3.4 约束路径推理算法

COPAR 模型获得无歧义约束推理路径的主要步骤包括主题实体及约束实体集区分、动作选择、实体选择、问题更新、实体及路径歧义约束处理. 具体如算法 1 所描述.

算法 1. COPAR 模型推理过程.

输入: 自然语言问题 q, 知识图谱 G, 主题实体 e_{te}, 约束实体集 E_c, 推理路径长度 L;

输出: 决策路径 h, 节点约束 c^*.

Begin

1. $e_0=e_{te}$, $h_0=\varnothing$, $c=[\cdot]*L$
2. $\boldsymbol{q}_0=QuestionEncoding(q)$
3. $s_0=(q,e_{te},e_0,h_0,c,\boldsymbol{q}_0)$
4. **For each** t in $[1,L]$:
5. $A(s_t)=\{r|(e_t,r)\in G\}$ //当前状态 s_t 的动作空间
6. **For** each $a=r$ in $A(s_t)$:
7. $\boldsymbol{a}=RelationEncoding(r)$
8. 计算动作 a 与问题 q 的语义分数 $S(a,q)$
9. **End for**
10. $\pi(a|s_t)=SoftMax(S(a,q))$ //对每个动作的得分归一化得到动作选择概率
11. 根据$\pi(a|s_t)$采样一个动作 $a_t=r_t$
12. $\boldsymbol{q}_t=UpdateQuestion(\boldsymbol{q}_t,\boldsymbol{r}_t)$ //更新问题表示
13. $E'=\{e|(e_t,r_t=a_t,e)\in G\}$ //得到关系 $r_t=a_t$ 对应的实体集 E'
14. 计算实体选择概率 P, 节点约束集 *constraint*
15. 根据实体选择概率 P, 选取实体 e_{t+1}, 确定节点约束 c_t
16. **For each** r_c, e_c in c_t:
17. $\boldsymbol{q}_t=UpdateQuestion(\boldsymbol{q}_t,\boldsymbol{r}_c)$ //更新问题表示
18. **End for**
19. $h_{t+1}=h_t+(a_t,e_{t+1})$ //更新决策路径
20. $c[t].append(c_t)$
21. $s_{t+1}=(q,e_{te},e_{t+1},h_{t+1},c,\boldsymbol{q}_t)$
22. **End for**
23. 对路径上的歧义约束进行处理得到 c^*
24. Return h, c^*

End

算法 1 的第 1 行对状态表示s_t中的部分元素进行了定义, 其中, e_0表示智能体最初所处的知识图谱中的节

点位置; 初始化为主题实体节点 e_{te}; h_0 表示历史决策路径, 用来记录推理路径选择的关系及实体; 初始化为空集∅; c 表示推理路径中每个节点的约束. 由于存在一个节点多个约束及有的节点无约束的情况, 因而容器 c 的大小设置为推理路径长度 L. 若实体选择过程选择的实体 e_{t+1} 有约束, 则添加至容器 c 的对应位置; 若无约束, 则对应位置为空. 第 2 行的函数 *QuestionEncoding*(·)采用第 3.1 节介绍的编码方法对问题 q 进行编码, 得到初始问题的向量化表示 $\boldsymbol{q}_0$. 进而, 第 3 行得到初始状态表示 $s_0=(q,e_{te},e_0,h_0,c,\boldsymbol{q}_0)$. 本文工作将推理路径设置为固定长度 L, 算法的第 4–22 行为模型每一跳所进行的关系及实体选择过程. 第 5–11 行为关系选择过程, 其中, 方法 *RelationEncoding*(·)对动作空间中每个动作 a 对应的关系 r 进行编码. 第 12 行的方法 *UpdateQuestion*(·)采用第 3.2.3 节介绍的方法针对选择的关系 r_t 对问题进行更新. 第 13–15 行为实体选择及约束获取过程. 第 16–18 行针对约束关系进行问题更新. 当选择完实体 e_{t+1} 后, 算法第 19 行将选择的关系及实体记录至容器 h_0. 第 20 行将加在该实体上的约束记录至容器 c, 相应地, 状态转移至 $s_{t+1}=(q,e_{te},e_{t+1},h_{t+1},c,\boldsymbol{q}_t)$. 重复第 5–21 行的动作及状态选择, 最终生成决策路径 $\tau=s_1,a_1,s_2,a_2,\ldots,s_L,a_L$ 及相应实体上的约束 c. 第 23 行对约束 c 存在路径上歧义的情况进行处理, 得到最终约束情况 c^*.

3.5 训练目标

在知识图谱问答任务中, 强化学习算法的训练目标为最大化在所有问答对上的折扣累计回报的期望, 如公式(3.9)所示:

$$J(\theta)=E_{(q,a)\in D}[E_{\tau\sim\pi_\theta(\tau)}[R(\tau)\,|\,(q,a)]] \tag{3.9}$$

$$R(\tau)=\sum_{t=0}^{\mathrm{T}}\gamma^t R^*(s_t,a_t,s_{t+1}) \tag{3.10}$$

其中, $(q,a)\in D$ 表示数据集 D 中的所有问答对(q,a), $\pi_\theta(\tau)$表示网络参数为θ时根据策略π产生路径τ, $R^*(\cdot)$为塑形奖励函数, γ为折扣因子.

相应地, 问答任务训练目标的策略梯度的形式化表示如公式(3.11)所示:

$$\nabla_\theta J(\theta)=E_{(q,a)\in D}\left[E_{\tau\sim\pi_\theta(\tau)}\left[\sum_{t=0}^{\mathrm{T}}\nabla_\theta\log\pi_\theta(a_t\,|\,s_t)\cdot\sum_{k=t}^{\mathrm{T}}R^*(s_k,a_k,s_{k+1})\right]\right] \tag{3.11}$$

其中, $\pi_\theta(a_t|s_t)$表示在状态 s_t 下根据策略π采取动作 a_t 的概率, $R(s_k,a_k,s_{k+1})$表示在状态 s_k 采取动作 a_k 到达状态 s_{k+1} 获得的奖励值. 在本文的约束问答任务中, 得到决策路径τ及最终约束 c 之后, 便可根据奖励函数计算方法, 计算每一跳决策的奖励值及折扣累计回报, 实现对当前策略网络参数θ进行更新.

4 实验分析

4.1 实验数据

本文主要在 4 个数据集上对我们提出的基于强化学习的知识图谱约束路径推理模型进行评估, 表 4.1 给出了数据集的统计细节, 其中, 第 2 列为知识图谱中实体或属性值个数, 第 3 列、第 4 列分别为关系个数和知识图谱中的三元组个数, 第 5 列为数据集各子集的问题数量, mix 表示数据集中所有问答对.

- PQ 和 PQL[22]: 即 PathQuestion 和 PathQuestion-Large, 这两个数据集是基于 Freebase 知识库构建的开放域知识图谱问答数据集. 其作者通过检索互联网和两个人类真实提出的数据集 WebQuestions[23]和 WikiAnswers[24], 对模板生成的问题进行同义词替换, 因而, 其中的自然语言问题的形式更加多样化, 句法结构和一些措辞也更加与真实世界的问题相似. 根据问题的推理路径长度, 这两个数据集均由 2 跳、3 跳的问答对数据构成;
- MetaQA: 2018 年, 由 Zhang[25]等人提出的电影领域的问答数据集. 该数据集中所有问题均依据 MovieQA 中的电影知识库 Wikidata 生成, 数据以问答对的形式呈现. 该数据集包含 3 个版本, Vanilla text data、NTM (neural translation model) text data 和 Audio data, 本文使用原始的 Vanilla text data 数据集. 该数据集根据其推理路径长度分为 1 跳、2 跳、3 跳的问答对数据, 本文主要在比较复杂的 2 跳、3 跳及混合的数据集上进行实验;

- KQA Pro: KQA Pro 将 Freebase 的子集 FB15k-237 中的实体对齐到 Wikidata 并对其进行扩展, 人工地将 Freebase 上的关系转换成 Wikidata 中的形式, 大大缩减了知识图谱的数据量, 且其问题种类更具多样性, 包含多种类型的问题. 本文主要研究其中带有实体或属性约束的多跳(multi-hop)问题, 通过其提供的 Find 及 FilterStr 功能及其他规则对问题进行筛选, 得到本文所使用的 KQA 子集 KQA-s. 为了更进一步地验证本文方法在约束问题上的有效性, 我们通过统计获取到问题的 Find 或 FilterStr 功能的个数, 将提取到的子集 KQA-s 划分为无约束子集 KQA-s-NC 和 KQA-s-C 这两个部分, 其中, KQA-s-NC 为 Find 或 FilterStr 功能个数为 1 的 KQA-s 子集, 包含了 KQA-s 中所有无约束多跳问题; KQA-s-C 为 Find 或 FilterStr 功能个数大于 1 的 KQA-s 子集, 包含了 KQA-s 中所有的约束多跳问题, KQA-s 即两者的混合.

表 4.1 实验数据集

数据集	实体数	关系数	三元组数	问题数	
PQ	2 215	14	4 049	2-hop 3-hop mix	1 908 5 198 7 106
PQL	5 035	364	12 248	2-hop 3-hop mix	1 594 1 031 2 625
MetaQA	43 234	9	134 741	2-hop 3-hop mix	44 610 42 671 122 018
KQA-s	131 890	960	489 224	mix con no-con	671 176 495

此外, 本文向知识图谱实体集合中的每个实体添加了一条指向节点自身的边. 另外, 为每个三元组添加其逆向边. 因此, 实际使用的知识图谱规模比原来的知识图谱要大. 实验过程中, 本研究将每个数据集按照 8:1:1 的比例随机分为训练集、验证集和测试集. 另外, 本实验在实际训练过程中, 将对每条问答对数据采样多次, 以扩充训练数据. 为了避免评估的偶然性, 本文使用了多个不同的随机因子来划分训练集、验证集及测试集, 下文报告的评估结果是重复实验多次后取的平均值.

4.2 评价指标及基准模型

本文仍采用智能体最终能否到达答案节点(即问答的准确率 Acc)来作为本文模型的评判标准, 准确率 Acc 的计算如公式(4.1)所示:

$$Acc = \frac{correct\_num}{N} \tag{4.1}$$

其中, *correct_num* 表示通过本研究提出的模型对测试集进行采样的推理路径最终到达答案节点的个数, N 表示总样本个数.

为了验证本文模型针对比较复杂的约束问题的有效性, 本文将与基于信息检索并取得重大成果的方法 KVMemNet、SRN、IRN 进行比较.

(1) KVMemNet[26]: 一个基于信息检索的典型模型. 将相关知识图谱子集以键值对的形式保存在内存中, 在寻找答案的过程中, 迭代地读取内存中的键值对来更新问题向量, 从而实现隐式推理. 对于比较复杂的问题, 需要更多的迭代次数, 键值记忆槽所占用的内存资源也更多;

(2) SRN[18]: 基于强化学习的框架进行逐跳推理模型. 该方法每次推理之前, 采用单层感知机对问题进行更新, 使用注意力机制使得模型在每一跳关注问题的不同部分. 但该方法将每次所能处理的信息限制为 1, 当推理过程中面临路径分支问题时, 该方法处理能力不强;

(3) IRN[27]: 具有可解释性的逐跳推理模型. 在推理过程中, 采用关系路径作为监督信息, 依据候选关系与问题的相似度选择每跳的关系, 并根据预测的关系对问题进行更新. 该方法采用词袋模型对问题进行编码, 导致语义信息有所缺失.

对于实验中涉及的超参数, 本文根据验证集上的结果进行手动调整. 在编码模块中, 本文将问题中词向量维度、动作空间中关系编码的词向量维度均设置为 100, 上述 3 个方法采用同样的词向量维度, 各方法的参数量见表 4.2. 模型采用已训练好的 Glove 词向量进行初始化, 编码层个数设置为 2, *dropout* 设为 0.1. 在约束选取部分, 三元组被认定为约束的条件是约束关系与当前问题的语义相似度计算大于阈值ε, 本文将阈值ε设置为 0. 在奖励函数设计及模型整体算法目标的计算中, 累计折扣回报的折扣因子γ设置为 0.95. 在整体算法训练过程中, 本文使用 ADAM 梯度优化器, 初始学习率 *lr* 设置为 0.000 1, *L*2 正则化权重因子λ设置为 0.5, 权重衰减系数μ设置为 0.01, 正则化损失β_e 设置为 0.01.

表 4.2 COPAR 与其他基准模型参数量对比(个)

模型	KVMemNet	IRN	SRN	COPAR
模型参数量	14 739 783	176 600	1 196 600	464 004

4.3 实验结果与分析

为了验证本文提出的基于强化学习的约束路径推理模型的通用性, 本节将所提出的模型与几个基于信息检索的先进方法在多跳数据集上的表现进行比较, 具体结果见表 4.3.

表 4.3 COPAR 与其他基准模型在不同数据集上的准确率比较(%)

	KVMemNet	IRN	SRN	COPAR
PQ-2H	91.5	96.4	96.3	98.4
PQ-3H	79.4	90.2	89.2	93.2
PQ-M	85.2	91.3	89.3	94.5
PQL-2H	70.5	83.8	78.6	89.6
PQL-3H	63.4	82.5	77.5	85.4
PQL-M	68.6	82.6	78.3	89.1
MetaQA-2H	84.3	95.5	95.1	97.2
MetaQA-3H	53.8	93.9	75.2	95.2
MetaQA-M	48.6	94.3	83.2	97.3

根据以上实验结果可以看出:

(1) 本研究提出的模型 COPAR 在 PathQuestion、PathQuestion-Large 及 MetaQA 数据集上优于当前先进的其他模型, 这表明本文所提模型能够有效地对特定领域及开放领域的多跳问题进行关系路径检测;

(2) 在关系数量相对较少的 PathQuestion 及 MetaQA 数据集上, COPAR 实现了较高的准确度. 本文认为, 这是仅采用当前实体连接的不重复出边作为动作空间, 从而减少了干扰项的原因. 在多跳问题上, 本文所提模型性能较优, 表明在跳数较多的情况下, 模型 COPAR 仍能有效地降低其他关系及实体的干扰;

(3) 在知识图谱规模较大且训练数据较少的数据集 PathQuestion-Large 上, COPAR 的性能与上述相对简单的数据集相比较低, 但优于其他模型, 我们将其原因归咎为本文所采用的问题实时更新及动作选择时的注意力机制.

KVMemNet 将与主题实体相关的整个子图考虑在内, 在知识图谱中, 即便关系数量少, 但每个实体节点的出度仍很大, 键值记忆槽很多, 干扰性较大, 且非常占用内存空间. IRN 虽有关系路径作为监督信息, 但其采用词袋模型对问题编码, 致使语义信息不全面. SRN 在每一步关系选择时, 将当前实体所连接的出边考虑在内, 与 KVMemNet 相比计算量减小, 但当实体出度较大时, 该方法仍会受到较多干扰. 本文所提方法只将当前实体所连接的不同关系考虑在内, 大大缩减了动作空间, 干扰项较少, 且本文针对带有实体或属性约束的问题进行了约束获取, 从实验结果可以看出, 本文所提方法明显优于上述模型.

为了验证本文提模型在带约束问题上的处理能力, 我们将数据集 KQA-s 划分为带约束和不带约束两个子集 KQA-s-C、KQA-s-NC, 分别在两个数据集上进行实验, 得到的各模型准确度见表 4.4.

表 4.4 所示结果表明, 本文所提方法 COPAR 在数据集 KQA-s 的准确度上具有绝对的优势, 验证了本文所

提方法在处理带有实体或属性约束问题上的有效性. 究其原因, 本文在带约束问题推理路径中一对多关系对应的实体上进行了约束判断, 准确定位到带约束实体, 提高了模型的整体准确度. 在不具有约束问题的数据集 KQA-s-NC 上, COPAR 与先进的基于图卷积网络的 RGCN 模型在准确率上表现相当, 验证了本文所提模型在大规模知识图谱上的路径推理能力. 在约束问题数据集 KQA-s-C 上, 本文所提模型远优于其他模型, RGCN 采用图卷积网络聚合当前实体邻域信息, 在获取到更多信息的同时, 干扰信息也更多. 总体而言, 该方法在一定程度上解决了带有实体或属性约束的问题, 但缺乏可解释性; SRN 采用强化学习框架, 其动作空间的设计使得该模型在每个时间步只选择实体的一条出边作为当前动作, 限制了其单个时间步的信息处理能力, 当遇到带约束问题使得推理路径产生分支时, 该方法在处理能力上表现出不足.

表 4.4 回答约束问题的准确率比较(%)

	KVMemNet	RGCN	SRN	COPAR	w/o con
KQA-s	35.8	66.4	55.2	71.6	64.2
KQA-s-NC	48.5	75.8	69.7	76.8	74.7
KQA-s-C	13.9	27.8	8.3	66.7	8.3

表 4.4 的最后一列 w/o con 表示在本文所提模型中不采用约束处理部分的实验结果. 可以看出: 当去掉约束获取部分后, 模型在约束问答数据集 KQA-s-C 上的准确率迅速下降, 与同样基于强化学习的方法 SRN 基本持平, 从而证明了对约束的处理在模型中解决约束问题时的重要性.

4.4 消融分析

为了进一步验证本研究所提模型中各部分设计的有效性, 本文针对各模块进行了消融实验, 通过实验验证各模块对整体模型的重要性.

(1) 基于注意力机制的动作选择策略的有效性验证

为了验证动作选择时注意力机制对模型整体性能的影响, 本文将动作选择时不采用注意力机制的模型 w/o attn 与整体模型 model 进行了对比, 在各混合数据集上的实验结果见表 4.5.

表 4.5 动作选择时注意力机制对整体模型的有效性验证(%)

	PQ-M	PQL-M	MetaQA-M	KQA-s
w/o attn	89.7	84.4	90.5	61.9
COPAR	94.9	89.1	97.6	71.7

实验结果表明: 与整体模型相比, 策略网络中动作选择时不采用注意力机制的模型在各混合数据集上的性能均有所下降. 本文分析, 造成这种结果的原因为: 在每一步关系选择时, 注意力机制往往能够关注到与当前问题最相关的信息, 一定程度上降低了其他信息的干扰. 实验表明了注意力机制在动作选择中的有效性.

(2) 约束处理部分的有效性验证

为了验证本文针对带有实体或属性约束问题提出的主题实体选取、基于约束获取的状态转移、歧义约束处理策略, 本文将整体模型 COPAR 与不采用约束处理(一切对约束的处理, 包括主题实体与约束实体集区分、约束获取及歧义约束处理)的模型 w/o CON、不采用主题实体与约束实体集区分部分的模型 w/o ENT 及不采用歧义约束处理部分的模型 w/o AMB 在带有约束问题的数据集 KQA-s 及 KQA-s-C 上进行了对比实验, 实验结果见表 4.6.

表 4.6 约束处理部分对整体模型的有效性验证(%)

	w/o ENT	w/o AMB	w/o CON	COPAR
KQA-s	66.4	70.1	64.2	71.7
KQA-s-C	47.2	61.1	8.3	66.7

表 4.6 所示的实验结果验证了本文所提出的对于约束的一系列处理措施的有效性. 在模型 w/o CON 上的实验结果表明, 本文的约束处理部分对解决带有约束的问题极为重要, 上一节已进行了分析, 下面主要分析主题实体选取及歧义约束处理对整体模型的影响.

在模型 w/o ENT 上的结果表明, 主题实体与约束实体集区分部分对模型的整体性能影响较大. 对于问句

中识别出来的多个实体或属性值, 主题实体选取部分中选择最优的实体作为主题实体, 一定程度上缓解了属性值作为主题实体造成推理路径变复杂或变长的情况.

在模型 w/o AMB 上的结果与整体模型相比准确度略低但相差不大. 造成该问题的原因可能是训练数据较少, 故而具有歧义的约束问题也相对较少. 但总体模型性能较优, 表明本文所提出的歧义约束处理策略是有效的.

(3) 实时问题更新的有效性验证

为了验证本文模型采用的实时问题更新部分的对于整体模型的作用, 我们在各数据集上仅对不采用问题更新部分的模型 w/o uq 进行了实验, 并与整体模型进行了对比, 实验结果见表 4.7.

表 4.7 实时问题更新部分对整体模型的有效性验证(%)

	PQ-M	PQL-M	MetaQA-M	KQA-s
w/o uq	85.6	73.2	76.8	41.0
COPAR	94.9	89.1	97.6	71.7

从实验结果中可以看出, 实时问题更新部分对整体模型性能影响较大. 这是由于在推理路径的中间节点处, 与该节点相关的关系可能会有多个, 通过注意力机制也无法完全区分当前最相关的关系. 而问题更新部分在每次选择完动作或约束之后, 都将其在问题中的相应部分进行掩盖, 在进行下一次选择时突出未处理过的相关关系, 提高了关系路径推理的准确度. 图 4.1 展示了在对问题“When was the film starred by Forest Whitaker and directed by Mark Rydell released ?”进行约束路径推理过程中的问题更新过程.

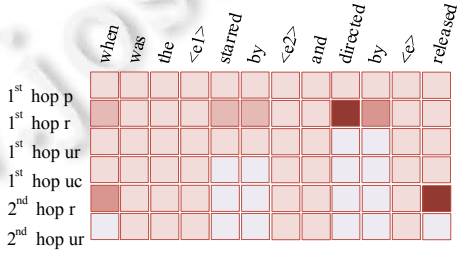


图 4.1 问题更新过程示例

图 4.1 所示的问题对应的路径为⟨Mark Rydell,directed_film,Even Money⟩, ⟨Even Money,released_time,2006-05-08⟩, 各实体对应约束为[·], [(instance of,film),(acted_film^{-1},Forest Whitaker)], [·]. 问题更新过程中, 在第 1 跳关系选择之前, 首先判断是否有约束作用在主题实体上, 若有, 则针对约束对应的关系对问题进行更新, 若无, 则忽略, 对应图中的 1st hop p; 当选择完第 1 跳关系 directed_film 之后, 将问题中与之相关的部分进行掩盖, 对应图中的 1st hop ur, 根据实体选择过程选择实体 Even Money, 并针对作用在该实体上的约束[(instance of,film),(acted_film^{-1},Forest Whitaker)]分别对问题进行更新, 对应图中的 1st hop uc, 从而在进行第 2 跳关系选择时, 模型大概率关注未处理的信息, 即 released_time 作为模型当前选择的关系, 对应图中的 2nd hop r, 从而实现逐步推理.

(4) 奖励函数的有效性验证

为了验证本文针对强化学习框架下多跳问答任务的奖励稀疏及延迟现象设计的基于潜力的塑形奖励函数的有效性, 本文仅对不采用塑形奖励模型 w/o pr 在各数据集上进行实验, 并与整体模型进行对比, 实验结果如图 4.2 所示.

图 4.2 所示的实验结果表明: 本文所设计的奖励函数在不改变智能体主线目标的情况下, 可以加速模型的收敛. 针对较长路径的奖励稀疏且延迟现象, 基于潜力的塑形奖励函数在每一次动作选择后都根据当前状态的潜力值返回奖励反馈, 若与上一状态相比潜力值变大, 则给一个正向奖励; 否则, 给一个负向奖励. 该方法使得模型的决策过程更具指导性, 促使模型保证在总体目标不变的情况下实现更早的收敛.

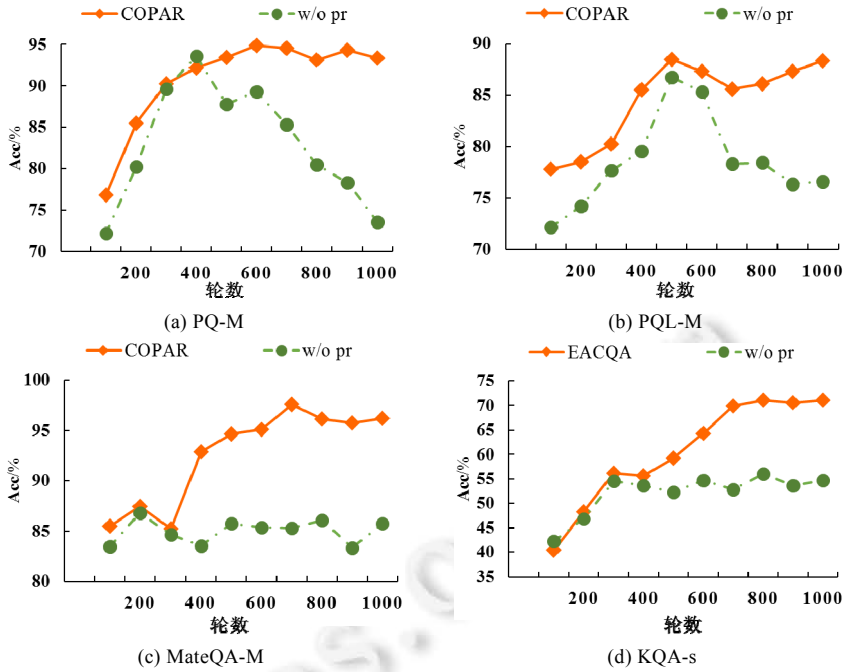


图 4.2 基于潜力的塑形奖励函数对整体模型的有效性验证

5 总 结

本文的知识图谱复杂问答推理建模为马尔可夫决策过程, 并提出一种基于强化学习的知识图谱约束问答推理模型 COPAR. 针对较长路径的奖励稀疏及延迟现象, 将约束路径与问题的相似度作为塑形奖励, 使得模型在整体训练目标不变的前提下, 可以更好地引导模型训练. 对于具有实体或属性约束的问题, 首先对主题实体、路径实体、约束实体进行区分, 然后对于路径推理过程中一个关系对应多个实体的情况, 提出了基于注意力机制的动作选择策略与基于约束的实体选择策略, 依据各实体满足约束的情况进行实体选择. 在每一次关系选择或约束确定之后, 根据已推理关系对问题嵌入表达进行动态更新, 从而保证推理路径的正确搜索方向. 此外, 设计了路径歧义处理策略, 进一步提升了路径推理的搜索效率. COPAR 模型打破了传统使用强化学习方法一跳只选取一个对应关系进行处理的限制, 尽可能多地处理当前可处理的信息, 缩短了推理路径, 提高了推理效率. 在多个基准数据集上进行了大量实验, 结果表明, COPAR 在多跳及约束问题上的处理能力均优于对比方法. 同时, 模型也可应用在多轮问答中, 通过将前一轮的问答作为下一轮的约束信息, 将多轮问答转换为约束问答.

References:

[1] Guan SP, Jin XL, Jia YT, Wang YZ, Cheng XQ. Knowledge reasoning over knowledge graph: A survey. Ruan Jian Xue Bao/Journal of Software, 2018, 29(10): 2966−2994 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5551.htm [doi: 10.13328/j.cnki. jos.005551]

[2] Wang X, Zou L, Wang ZK, Peng P, Feng ZY. Research on knowledge graph data management: A survey. Ruan Jian Xue Bao/Journal of Software, 2019, 30(7): 2139−2174 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5841.htm [doi: 10.13328/j.cnki.jos.005841]

[3] Liu Z, Liu C, Lin W, Zhao J. Pretraining financial language model with multi-task learning for financial text mining. Journal of Computer Research and Development, 2021, 58(8): 1761−1772 (in Chinese with English abstract).

[4] Zhou M, Peng S, Yang M, Li N, Wang H, Qiao L, Mi HT, Wen ZJ, Xu T, Liu, L. IIAS: An intelligent insurance assessment system through online real-time conversation analysis. In: Zhou H, ed. Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence. Montreal: ijcai.org, 2021. 5036−5039. [doi: 10.24963/ijcai.2021/721]

[5] Zhang YJ, Dong Z, Meng XW. Pesearch on personalized advertising pecommendation systems and their applications. Chinese Journal of Computers, 2021, 44(3): 531−563 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2021.20210298]

[6] Zheng YZ, Zhu DJ, Wu HL, Peng XR. Overview on knowledge graph question answering. Computer Systems & Applications, 2022, 31(4): 1−13 (in Chinese with English abstract). [doi: 10.15888/j.cnki.csa.008418]

[7] Bi X, Nie H, Zhang XY, Zhao XG, Yuan Y, Wang GR. Unrestricted multi-hop reasoning network for interpretable question answering over knowledge graph. Knowledge-based Systems, 2022, 243: 108515. [doi: 10.1016/j.knosys.2022.108515]

[8] Sun YW, Cheng G, Li X, Qu YZ. Graph matching network for interpretable complex question answering over knowledge graphs. Journal of Computer Research and Development, 2021, 58(12): 2673−2683 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2021.20211004]

[9] Zhu AJ, Ouyang DQ, Liang S, Shao J. Step by step: A hierarchical framework for multi-hop knowledge graph reasoning with reinforcement learning. Knowledge-based Systems, 2022, 248: 108843. [doi: 10.1016/j.knosys.2022.108843]

[10] Kaiser M, Saha Roy R, Weikum G. Reinforcement learning from reformulations in conversational question answering over knowledge graphs. In: Diaz F, ed. Proc. of the 44th Int'l Conf. on Research and Development in Information Retrieval. ACM, 2021. 459−469. [doi: 10.1145/3404835.3462859]

[11] Gardner M, Talukdar PP, Kisiel B, Mitchell TM. Improving learning and inference in a large knowledge-base using latent syntactic cues. In: Yarowsky D, ed. Proc. of the 2013 Conf. on Empirical Methods in Natural Language Processing. Seattle: Association for Computational Linguistics, 2013. 833−838.

[12] Xiong W, Hoang T, Wang WY. Deeppath: A reinforcement learning method for knowledge graph reasoning. In: Palmer M, ed. Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. 564−573. [doi: 10.18653/v1/d17-1060]

[13] Das R, Dhuliawala S, Zaheer M, Vilnis L, Durugkar I, Krishnamurthy A, Smola A, McCallum A. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[14] Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 1992, 8(3−4): 229−256.

[15] Shen Y, Chen J, Huang PS, Guo YQ, Gao JF. M-Walk: Learning to walk over graphs using Monte Carlo tree search. In: Bengio S, ed. Proc. of the Advances in Neural Information Processing Systems 31: Annual Conf. on Neural Information Processing Systems. Montréal, 2018. 6787−6798.

[16] Wan G, Pan S, Gong C, Zhou C, Haffari G. Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning. In: Bessiere C, ed. Proc. of the 29th Int'l Joint Conf. on Artificial Intelligence. 2020. 1926−1932. [doi: 10.24963/ijcai.2020/267]

[17] Wang Q, Hao Y, Cao J. ADRL: An attention-based deep reinforcement learning framework for knowledge graph reasoning. Knowledge-based Systems, 2020, 197: 105910. [doi: 10.1016/j.knosys.2020.105910]

[18] Zhang Q, Wang X, Zhou G, Zhang Y, Jimmy Huang JX. ARL: An adaptive reinforcement learning framework for complex question answering over knowledge base. Information Processing & Management, 2022, 59(3): 102933. [doi: 10.1016/j.ipm.2022.102933]

[19] Qiu Y, Wang Y, Jin X, Zhang K. Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In: Caverlee J, ed. Proc. of the 13th Association for Computing Machinery (ACM) Int'l Conf. on Web Search and Data Mining. Houston: ACM, 2020. 474−482. [doi: 10.1145/3336191.3371812]

[20] Lin XV, Socher R, Xiong C. Multi-hop knowledge graph reasoning with reward shaping. In: Riloff E, ed. Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 3243−3253. [doi: 10.18653/v1/d18-1362]

[21] Kaiser M, Roy RS, Weikum G. Reinforcement learning from reformulations in conversational question answering over knowledge graphs. In: Diaz F, ed. Proc. of the 44th Int'l Association for Computing Machinery (ACM) Conf. on Research and Development in Information Retrieval. ACM, 2021. 459−469. [doi: 10.1145/3404835.3462859]

[22] Ng AY, Harada D, Russell SJ. Policy invariance under reward transformations: Theory and application to reward shaping. In: Ivan B, ed. Proc. of the 16th Int'l Conf. on Machine Learning. Bled: Morgan Kaufmann Publishers, 1999. 278−287.

[23] Kitaev N, Klein D. Constituency parsing with a self-attentive encoder. In: Gurevych I, ed. Proc. of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: Association for Computational Linguistics, 2018. 2676−2686. [doi: 10.18653/v1/P18-1249]

[24] Zhou M, Huang M, Zhu X. An interpretable reasoning network for multi-relation question answering. In: Bender EM, ed. Proc. of the 27th Int'l Conf. on Computational Linguistics. Santa Fe: Association for Computational Linguistics, 2018. 2010−2022.

[25] Berant J, Chou A, Frostig R, Liang P. Semantic parsing on freebase from question-answer pairs. In: Yarowsky D, ed. Proc. of the 2013 Conf. on Empirical Methods in Natural Language Processing. Seattle: Association for Computational Linguistics, 2013. 1533−1544.

[26] Fader A, Zettlemoyer L, Etzioni O. Paraphrase-driven learning for open question answering. In: Schuetze H, ed. Proc. of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia: Association for Computational Linguistics, 2013. 1608−1618.

[27] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: Burstein J, ed. Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2019. 4171−4186. [doi: 10.18653/v1/n19-1423]

[28] Miller A, Fisch A, Dodge J, Karimi AH, Bordes A, Weston J. Key-value memory networks for directly reading documents. In: Su J, ed. Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing. Austin: Association for Computational Linguistics, 2016. 1400−1409. [doi: 10.18653/v1/d16-1147]

[29] Zhou M, Huang M, Zhu X. An interpretable reasoning network for multi-relation question answering. In: Bender EM, ed. Proc. of the 27th Int'l Conf. on Computational Linguistics. Santa Fe: Association for Computational Linguistics, 2018. 2010−2022.

附中文参考文献:

[1] 官赛萍, 靳小龙, 贾岩涛, 王元卓, 程学旗. 面向知识图谱的知识推理研究进展. 软件学报, 2018, 29(10): 2966−2994. http://www.jos.org.cn/1000-9825/5551.htm [doi: 10.13328/j.cnki.jos.005551]

[2] 王鑫, 邹磊, 王朝坤, 彭鹏, 冯志勇. 知识图谱数据管理研究综述. 软件学报, 2019, 30(7): 2139−2174. http://www.jos.org.cn/1000-9825/5841.htm [doi:10.13328/j.cnki.jos.005841]

[3] 刘壮, 刘畅, Wayne Lin, 赵军. 用于金融文本挖掘的多任务学习预训练金融语言模型. 计算机研究与发展, 2021, 58(8): 1761−1772.

[5] 张玉洁, 董政, 孟祥武. 个性化广告推荐系统及其应用研究. 计算机学报, 2021, 44(3): 531−563. [doi: 10.7544/issn1000-1239.2021.20210298]

[6] 郑泳智, 朱定局, 吴惠粦, 彭小荣. 知识图谱问答领域综述. 计算机系统应用, 2022, 31(4): 1−13. [doi: 10.15888/j.cnki.csa.008418]

[8] 孙亚伟, 程龚, 厉肖, 瞿裕忠. 基于图匹配网络的可解释知识图谱复杂问答方法. 计算机研究与发展, 2021, 58(12): 2673−2683. [doi: 10.7544/issn1000-1239.2021.20211004]

毕鑫(1987－), 男, 博士, 副研究员, CCF 专业会员, 主要研究领域为大数据管理与分析, 知识图谱, 半结构化数据管理.

聂豪杰(1996－), 男, 博士生, CCF 学生会员, 主要研究领域为机器学习, 知识图谱.

赵相国(1973－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为大数据管理与分析, 智能分析与决策, 深度学习.

袁野(1981－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为大数据管理, 数据库理论与系统.

王国仁(1965－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为不确定数据管理, 数据密集型计算, 可视媒体数据分析管理, 非结构化数据管理, 分布式查询处理与优化, 生物信息学.