THINK-ON-GRAPH: DEEP AND RESPONSIBLE REASONING OF LARGE LANGUAGE MODEL ON KNOWLEDGE GRAPH

Jiashuo Sun[21*†] **Chengjin Xu**[1*] **Lumingyuan Tang**[31*†] **Saizhuo Wang**[41*†]
Chen Lin[2] **Yeyun Gong**[6] **Lionel M. Ni**[5] **Heung-Yeung Shum**[14] **Jian Guo**[15‡]
[1]IDEA Research, International Digital Economy Academy
[2]Xiamen University
[3]University of Southern California
[4]The Hong Kong University of Science and Technology
[5]The Hong Kong University of Science and Technology (Guangzhou)
[6]Microsoft Research Asia

ABSTRACT

Although large language models (LLMs) have achieved significant success in various tasks, they often struggle with hallucination problems, especially in scenarios requiring deep and responsible reasoning. These issues could be partially addressed by introducing external knowledge graphs (KG) in LLM reasoning. In this paper, we propose a new LLM-KG integrating paradigm "LLM ⊗ KG" which treats the LLM as an agent to interactively explore related entities and relations on KGs and perform reasoning based on the retrieved knowledge. We further implement this paradigm by introducing a new approach called Think-on-Graph (ToG), in which the LLM agent iteratively executes beam search on KG, discovers the most promising reasoning paths, and returns the most likely reasoning results. We use a number of well-designed experiments to examine and illustrate the following advantages of ToG: 1) compared with LLMs, ToG has better deep reasoning power; 2) ToG has the ability of knowledge traceability and knowledge correctability by leveraging LLMs reasoning and expert feedback; 3) ToG provides a flexible plug-and-play framework for different LLMs, KGs and prompting strategies without any additional training cost; 4) the performance of ToG with small LLM models could exceed large LLM such as GPT-4 in certain scenarios and this reduces the cost of LLM deployment and application. As a training-free method with lower computational cost and better generality, ToG achieves overall SOTA in 6 out of 9 datasets where most previous SOTAs rely on additional training. Our code is publicly available at `https://github.com/IDEA-FinAI/ToG`.

*Equal contribution.
†Work done during internship at IDEA Research.
‡Corresponding author.

Published as a conference paper at ICLR 2024

THINK-ON-GRAPH: DEEP AND RESPONSIBLE REASONING OF LARGE LANGUAGE MODEL ON KNOWLEDGE GRAPH

Jiashuo Sun[21*†] **Chengjin Xu**[1*] **Lumingyuan Tang**[31*†] **Saizhuo Wang**[41*†]
Chen Lin[2] **Yeyun Gong**[6] **Lionel M. Ni**[5] **Heung-Yeung Shum**[14] **Jian Guo**[15‡]
[1]IDEA Research, International Digital Economy Academy
[2]Xiamen University
[3]University of Southern California
[4]The Hong Kong University of Science and Technology
[5]The Hong Kong University of Science and Technology (Guangzhou)
[6]Microsoft Research Asia

ABSTRACT

虽然大型语言模型（LLMs）在各种任务中取得了显著成功，但它们在需要深度和负责任推理的场景中往往会遇到幻觉问题。这些问题可以通过在LLM推理中引入外部知识图（KG）部分解决。本文提出了一种新的LLM-KG集成范式“LLM ⊗ KG”，将LLM视为一个智能体（agent），以交互方式探索KG上的相关实体和关系，并基于检索到的知识进行推理。我们进一步通过引入一种新的方法称为图上思维（Think-on-Graph, ToG）来实现这一范式，在该方法中，LLM智能体迭代执行KG上的束搜索，发现最有前途的推理路径，并返回最可能的推理结果。我们利用一系列设计良好的实验来检验和说明ToG的以下优势：1）与LLM相比，ToG具有更好的深度推理能力；2）ToG通过利用LLMs的推理和专家反馈，具备知识可追溯性和知识可修正性；3）ToG为不同的LLM、KG和提示策略提供了灵活的插拔式框架，而无需任何额外的训练成本；4）在某些场景下，小型LLM模型的ToG性能超过大型LLM，如GPT-4，从而降低了LLM部署和应用的成本。作为一种无训练方法，ToG具有较低的计算成本和更好的通用性，在9个数据集中的6个上达成了整体SOTA，而大多数之前的SOTA依赖于额外的训练。我们的代码可在`https://github.com/IDEA-FinAI/ToG`公开获取。

*Equal contribution.
†Work done during internship at IDEA Research.
‡Corresponding author.

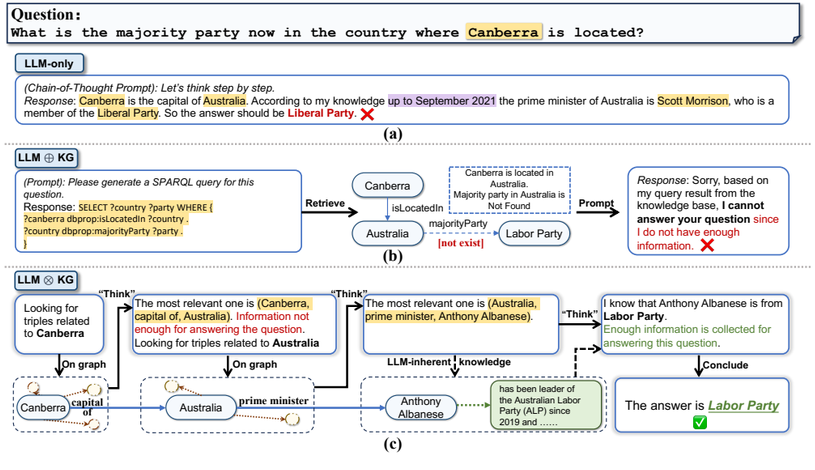

图 1: Representative workflow of three LLM reasoning paradigms: (a) LLM-only (e.g., Chain-of-Thought prompting), (b) LLM ⊕ KG (e.g., KBQA via LLM-generated SPARQL query), (c) LLM ⊗ KG (e.g., Think-on-Graph).

1 INTRODUCTION

Large language models (LLMs) (Ouyang et al., 2022; OpenAI, 2023; Thoppilan et al., 2022; Brown et al., 2020a; Chowdhery et al., 2022; Touvron et al., 2023) have demonstrated remarkable performance across various natural language processing tasks. These models capitalize on pre-training techniques applied to vast text corpora to generate responses that are coherent and contextually appropriate. Despite their impressive performance, LLMs have substantial limitations when facing complex knowledge reasoning tasks (Petroni et al., 2021; Talmor et al., 2019; Talmor & Berant, 2018; Zhang et al., 2023) that require deep and responsible reasoning. Firstly, LLMs usually fail to provide accurate answers to questions requiring specialized knowledge beyond what was included in the pre-training phase (out-of-date knowledge in Figure 1a), or to questions requiring long logic chain and multi-hop knowledge reasoning. Secondly, LLMs lack responsibility, explainability and transparency, raising concerns about the risk of hallucinations or toxic texts. Thirdly, the training process for LLMs is often expensive and time-consuming, making it challenging to keep their knowledge up to date.

Recognizing these challenges, a natural and promising solution is to incorporate external knowledge such as knowledge graphs (KGs) to help improve LLM reasoning. KGs offer structured, explicit, and editable representations of knowledge, presenting a complementary strategy to mitigate the limitations of LLMs (Pan et al., 2023). Researchers (Li et al., 2023c; Xie et al., 2022; Baek et al., 2023b; Yang et al., 2023; Wang et al., 2023a; Jiang et al., 2023) have explored the usage of KGs as external knowledge sources to mitigate hallucination in LLMs. These approaches follow a routine: retrieve information from KGs, augment the prompt accordingly, and feed the increased prompt into LLMs (as illustrated in Figure 1b). In this paper, we refer to this paradigm as "LLM⊕KG". Although aiming to integrate the power of LLM and KG, in this paradigm, LLM plays the role of translator which transfers input questions to machine-understandable command for KG searching and reasoning, but it does not participate in the graph reasoning process directly. Unfortunately, the loose-coupling LLM ⊕ KG paradigm has its own limitations, and its success depends heavily on the completeness

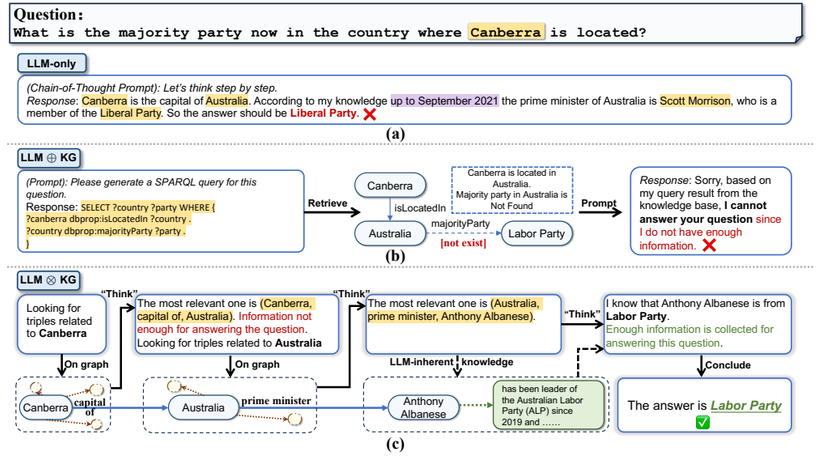

图 1: 三种大型语言模型（LLM）推理范式的代表性工作流程：(a) 仅使用LLM（例如，思维链提示），(b) LLM ⊕ 知识图谱（KG）（例如，通过LLM生成的SPARQL查询进行知识库问答（KBQA）），(c) LLM ⊗ 知识图谱（KG）（例如，图上思考）。

1 INTRODUCTION

大型语言模型（LLMs）(Ouyang et al., 2022; OpenAI, 2023; Thoppilan et al., 2022; Brown et al., 2020a; Chowdhery et al., 2022; Touvron et al., 2023) 在各种自然语言处理任务中展示了卓越的性能。这些模型利用应用于大量文本语料库的预训练技术，生成连贯且符合上下文的回应。尽管它们表现出色，LLMs 在面对复杂的知识推理任务时仍有显著的局限性(Petroni et al., 2021; Talmor et al., 2019; Talmor & Berant, 2018; Zhang et al., 2023)，这些任务需要深入且负责任的推理。首先，LLMs 通常无法提供准确的答案，对那些需要超出预训练阶段所包含的专门知识的问题（如图 1a 中的过时知识）或需要长逻辑链和多跳知识推理的问题。其次，LLMs 缺乏责任感、可解释性和透明度，这引发了对幻觉或有害文本风险的担忧。第三，LLMs 的训练过程通常成本高昂且耗时，使得保持其知识的更新变得具有挑战性。

认识到这些挑战，一个自然且有前景的解决方案是结合外部知识，如知识图谱（KGs），以帮助改善大语言模型（LLM）的推理能力。知识图谱提供了结构化、明确且可编辑的知识表征（representation），提供了一种补充策略以减轻大语言模型的局限性 (Pan et al., 2023)。研究人员 (Li et al., 2023c; Xie et al., 2022; Baek et al., 2023b; Yang et al., 2023; Wang et al., 2023a; Jiang et al., 2023) 探索了将知识图谱作为外部知识源的使用以减轻大语言模型中的幻觉。这些方法遵循一个常规：从知识图谱中检索信息，相应地增强提示，并将增强后的提示输入到大语言模型中（如图 1b 所示）。在本文中，我们将这一范式称为 "LLM ⊕ KG"。虽然旨在整合大语言模型和知识图谱的力量，但在这一范式中，大语言模型充当翻译者，将输入问题转换为机器可理解的命令以进行知识图谱搜索和推理，但并不直接参与图的推理过程。不幸的是，松耦合的 "LLM ⊕ KG" 范式有其自身的局限性，其成功在很大程度上依赖于知识图谱的完整性和高质量。例如，在图 1b 中，尽管大语言模型成功识别了回答问题所需的必要关系类型，但缺少 "主要政党" 这一关系导致未能检索出正确的答案。

基于这些考虑，我们提出了一种新的紧耦合 "LLM⊗KG" 范式，其中知识图谱与大语言模型协同工作，在每一步图推理中互补各自的能力。图 1c 提供了一个示例，说明了 "LLM⊗KG"

and high quality of KG. In Figure 1b, for example, although LLM successfully identified necessary relation types required to answer the question, the absence of the relation “majority party” leads to a failure in retrieving the correct answer.

Building upon these considerations, we propose a new tight-coupling “LLM $\otimes$ KG” paradigm where KGs and LLMs work in tandem, complementing each other’s capabilities in each step of graph reasoning. Figure 1c provides an example illustrating the advantage of LLM $\otimes$ KG. In this example, the missing relation "majority party" resulting in the failure in Figure 1b can be complemented by a reference triple (**Australia**, **prime minister**, **Anthony Albanese**) discovered by the LLM agent with dynamic reasoning ability (Yao et al., 2022), as well as the political party membership of **Anthony Albanese** coming from LLM’s inherent knowledge. In this way, the LLM succeeds in generating the correct answer with reliable knowledge retrieved from KGs. As an implementation of this paradigm, we propose an algorithmic framework “Think-on-Graph” (meaning: LLMs “Think” along the reasoning paths “on” knowledge “graph” step-by-step, abbreviated as ToG below), for deep, responsible, and efficient LLM reasoning. Using the beam search algorithm (Jurafsky & Martin, 2009) in KG/LLM reasoning (Atif et al., 2023; Sun et al., 2023a; Xie et al., 2023; Liu et al., 2024), ToG allows LLM to dynamically explore a number of reasoning paths in KG and make decisions accordingly. Given an input question, ToG first identifies initial entities and then iteratively calls the LLM to retrieve relevant triples from KGs through exploration (looking for relevant triples in KG via “on graph” step) and reasoning (deciding on the most relevant triples via “think” step) until adequate information through the top-N reasoning paths in beam search is gathered to answer the question (judged by LLMs in "Think" step) or the predefined maximum search depth is reached.

The advantage of ToG can be abbreviated as (1) **Deep reasoning:** ToG extracts diverse and multi-hop reasoning paths from KGs as the basis for LLM reasoning, enhancing LLMs’ deep reasoning capabilities for knowledge-intensive tasks. (2) **Responsible reasoning:** Explicit, editable reasoning paths improve the explainability of the reasoning process of LLMs, and enable the tracing and correction of the provenances of models’ outputs. (3) **Flexibility and efficiency:** a) ToG is a plug-and-play framework that can be applied to a variety of LLMs and KGs seamlessly. b) Under ToG framework, knowledge can be updated frequently via KG instead of LLM whose knowledge-update is expensive and slow. c) ToG enhances the reasoning ability of small LLMs (e.g., LLAMA2-70B) to be competitive with big LLMs (e.g., GPT-4).

2 METHODS

ToG implements the “LLM $\otimes$ KG” paradigm by asking LLM to perform beam search on knowledge graph. Specifically, it prompts the LLM to iteratively explore multiple possible reasoning paths on KGs until the LLM determines that the question can be answered based on the current reasoning paths. ToG constantly updates and maintains top-N reasoning paths $P = \{p_1, p_2, \ldots, p_N\}$ for the question x after each iteration, where N denotes the width of beam search. The entire inference process of ToG contains the following 3 phases: initialization, exploration, and reasoning.

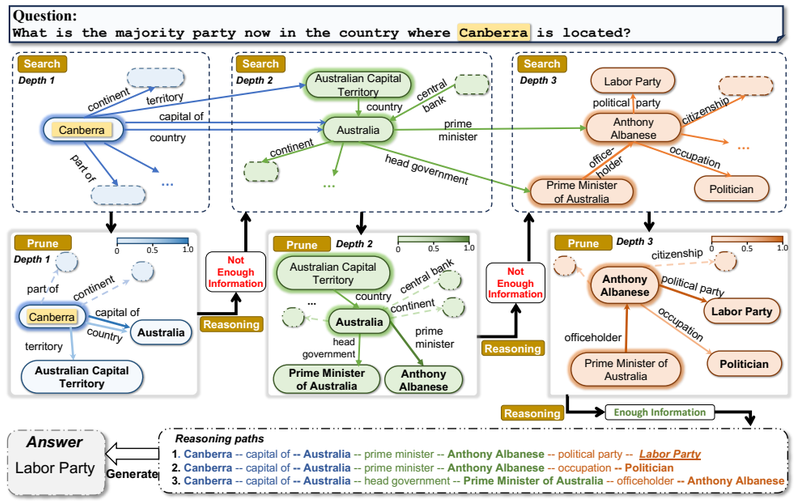

图 2: 一个 ToG 的示例工作流程。发光的实体是每次迭代（深度）开始时的中心实体，而 **粗体** 的实体是经过修剪后为下一次迭代选定的中心实体。在每个修剪步骤中，边缘的暗度代表 LLM 给出的排名分数，虚线表示由于评估分数低而被修剪的关系。

的优势。在这个例子中，导致图 1b 中失败的缺失关系 "主要政党" 可以通过大语言模型智能体使用动态推理能力发现的引用三元组 (**Australia**, **prime minister**, **Anthony Albanese**) 来补充，以及 **Anthony Albanese** 的政党成员身份来源于大语言模型的内在知识。通过这种方式，大语言模型成功地生成了利用知识图谱检索的可靠知识得出的正确答案。作为该范式的实现，我们提出了一个算法框架 “Think-on-Graph”（意味着：大语言模型在知识 “图” 上沿推理路径 “思考”，一步一步进行，简称为 ToG），以进行深度、负责任和高效的大语言模型推理。使用束搜索算法 (Jurafsky & Martin, 2009) 在知识图谱/大语言模型推理 (Atif et al., 2023; Sun et al., 2023a; Xie et al., 2023; Liu et al., 2024) 中，ToG 允许大语言模型动态探索知识图谱中的多个推理路径并据此做出决策。给定一个输入问题，ToG 首先识别初始实体，然后通过探索（在知识图谱中寻找相关三元组的 “在图上” 步骤）和推理（通过 “思考” 步骤决定最相关的三元组）迭代调用大语言模型以检索知识图谱中的相关三元组，直到通过束搜索获取足够的信息，以回答问题（由大语言模型在 “思考” 步骤中判断）或达到预定义的最大搜索深度。

ToG 的优势可以概括为 (1) **深度推理:** ToG 从知识图谱中提取多样的多跳推理路径，作为大语言模型推理的基础，增强了大语言模型在知识密集型任务中的深度推理能力。 (2) **负责任推理:** 明确、可编辑的推理路径提高了大语言模型推理过程的可解释性，并能追踪和纠正模型输出的来源。 (3) **灵活性和效率:** a) ToG 是一个即插即用的框架，可以无缝地应用于各种大语言模型和知识图谱。 b) 在 ToG 框架下，知识可以通过知识图谱频繁更新，而不是大语言模型，其知识更新昂贵且缓慢。 c) ToG 增强了小型大语言模型（例如，LLAMA2-70B）的推理能力，使其能够与大型大语言模型（例如，GPT-4）竞争。

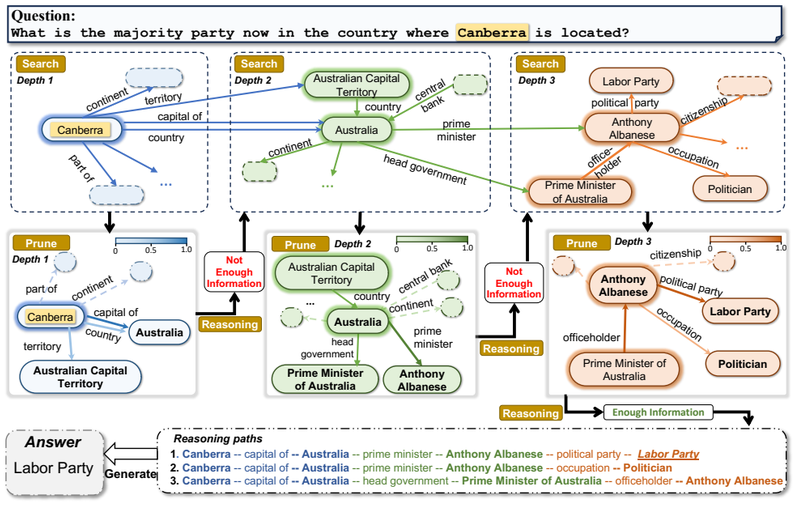

图 2: An example workflow of ToG. The glowing entities are the central entities where the search starts at each iteration (depth), and the entities with **boldface** are the selected central entities for the next iteration after pruning. At each pruning step, the darkness of the edges represents the ranking score given by LLM, and the dashed lines indicate relations that have been pruned due to low evaluation scores.

2.1 THINK-ON-GRAPH

2.1.1 INITIALIZATION OF GRAPH SEARCH

Given a question, ToG leverages the underlying LLM to localize the initial entity of the reasoning paths on knowledge graph. This phase can be regarded as the initialization of the top-N reasoning paths P. ToG first prompts LLMs to automatically extract the topic entities in question and gets the top-N topic entities $E^0 = \{e_1^0, e_2^0, ..., e_N^0\}$ to the question. Note that the number of topic entities might possibly be less than N.

2.1.2 EXPLORATION

At the beginning of the D-th iteration, each path p_n consists of $D-1$ triples, i.e., $p_n = \{(e_{s,n}^d, r_{j,n}^d, e_{o,n}^d)\}_{d=1}^{D-1}$, where $e_{s,n}^d$ and $e_{o,n}^d$ denote subject and object entities, $r_{j,n}^d$ is a specific relation between them, $(e_{s,n}^d, r_{j,n}^d, e_{o,n}^d)$ and $(e_{s,n}^{d+1}, r_{j,n}^{d+1}, e_{o,n}^{d+1})$ are connected to each other. The sets of the tail entities and relations in P are denoted as $E^{D-1} = \{e_1^{D-1}, e_2^{D-1}, ..., e_N^{D-1}\}$ and $R^{D-1} = \{r_1^{D-1}, r_2^{D-1}, ..., r_N^{D-1}\}$, respectively.

The exploration phase in the D-th iteration aims to exploit the LLM to identify the most relevant top-N entities E^D from the neighboring entities of the current top-N entity set E^{D-1} based on the question x and extend the top-N reasoning paths P with E^D. To address the complexity of handling numerous neighboring entities with the LLM, we implement a two-step exploration strategy: first, exploring significant relations, and then using selected relations to guide entity exploration.

2 METHODS

ToG 实现了 “LLM ⊗ KG” 范式，通过要求 LLM 在知识图谱上执行束搜索。具体而言，它提示 LLM 迭代地探索多个可能的推理路径，直到 LLM 确定该问题可以基于当前的推理路径得到回答。ToG 在每次迭代后不断更新和维护与问题 x 相关的前 N 条推理路径 $P = \{p_1, p_2, \ldots, p_N\}$，其中 N 表示束搜索的宽度。 ToG 的整个推理过程包含以下 3 个阶段：初始化、探索（exploration）和推理。

2.1 THINK-ON-GRAPH

2.1.1 INITIALIZATION OF GRAPH SEARCH

给定一个问题，ToG 利用基础的 LLM（大型语言模型）来定位知识图谱上的初始实体，这一阶段可以视为顶级 N 推理路径 P 的初始化。ToG 首先提示 LLM 自动提取问题中的主题实体，并获取顶级 N 主题实体 $E^0 = \{e_1^0, e_2^0, ..., e_N^0\}$。请注意，主题实体的数量可能少于 N。

2.1.2 EXPLORATION

在第 D 次迭代开始时，每条路径 p_n 由 $D-1$ 个三元组组成，即 $p_n = \{(e_{s,n}^d, r_{j,n}^d, e_{o,n}^d)\}_{d=1}^{D-1}$，其中 $e_{s,n}^d$ 和 $e_{o,n}^d$ 分别表示主题和客体实体，$r_{j,n}^d$ 是它们之间的特定关系，$(e_{s,n}^d, r_{j,n}^d, e_{o,n}^d)$ 和 $(e_{s,n}^{d+1}, r_{j,n}^{d+1}, e_{o,n}^{d+1})$ 彼此相连。尾实体和关系的集合 P 被表示为 $E^{D-1} = \{e_1^{D-1}, e_2^{D-1}, ..., e_N^{D-1}\}$ 和 $R^{D-1} = \{r_1^{D-1}, r_2^{D-1}, ..., r_N^{D-1}\}$，分别。

第 D 次迭代中的探索阶段旨在利用大语言模型（LLM）根据问题 x 识别当前顶层 N 实体集合 E^{D-1} 的邻近实体中最相关的顶层 N 实体 E^D，并扩展顶层 N 推理路径 P 与 E^D。为了应对使用 LLM 处理大量邻近实体的复杂性，我们实施了双步骤探索策略：首先，探索重要关系，然后使用选定的关系来引导实体探索。

关系探索 关系探索（relation exploration）是一个从 E^{D-1} 到 R^D 的深度为 1、宽度为 N 的广度优先搜索过程。整个过程可以分解为两个步骤：搜索（Search）和 剪枝（Prune）。大语言模型（LLM）作为智能体（agent）自动完成这个过程。

- **搜索** 在第 D 次迭代的开始阶段，关系探索（exploration）阶段首先针对每条推理路径 p_n 搜索与尾实体 e_n^{D-1} 关联的关系 $R_{cand,n}^D$。这些关系被汇总为 R_{cand}^D。在图 2 的情况下，$E^1 = \{\textbf{Canberra}\}$，$R_{cand}^1$ 表示与 **Canberra** 内部或外部链接的所有关系的集合。值得注意的是，搜索 程序可以通过执行附录 E.1 和 E.2 中展示的两个简单的预定义查询轻松完成，这使得 ToG 能够很好地适应不同的知识图谱（KG），**而无需任何训练成本**。
- **剪枝** 一旦我们从关系搜索中获得了候选关系集 R_{cand}^D 和扩展的候选推理路径 P_{cand}，我们可以利用 LLM 选择出新的 top-N 推理路径 P，这些路径以尾关系 R^D 结尾，基于问题 x 的字面信息和候选关系 R_{cand}^D。这里使用的提示可以在附录 E.3.1 中找到。如图 2 所示，LLM 在第一次迭代中从所有与实体 **堪培拉**（Canberra）相关的关系中选出了 top-3 关系 {**首都属于（capital of）**, **国家（country）**, **领土（territory）**}。由于 **堪培拉** 是唯一的主题实体，top-3 候选推理路径更新为 {(**堪培拉, 首都属于（capital of**), (**堪培拉, 国家（country）**),(**堪培拉, 领土（territory）**)}。

Entity Exploration 与关系探索（exploration）类似，实体探索（entity exploration）也是一个由LLM从 R^D 到 E^D 执行的启发式搜索过程，包含两个步骤，`Search` 和 `Prune`。

Relation Exploration Relation exploration is a beam search process with the depth of 1 and the width of N from E^{D-1} to R^D. The whole process can be decomposed into two steps: `Search` and `Prune`. The LLM serves as an agent to automatically complete this process.

- **Search** At the beginning of the D-th iteration, the relation exploration phase first searches out relations $R^D_{cand,n}$ linked to the tail entity e_n^{D-1} for each reasoning path p_n. These relations are aggregated into R^D_{cand}. In the case of Figure 2, $E^1 = \{\textbf{Canberra}\}$ and R^1_{cand} denotes the set of all relations linked to **Canberra** inwards or outwards. Notably, the `Search` procedure can be easily completed by executing two simple pre-defined formal queries shown in Appendix E.1 and E.2, which makes ToG adapt well to different KGs **without any training cost**.
- **Prune** Once we have obtained the candidate relation sets R^D_{cand} and the expanded candidate reasoning paths P_{cand} from the relation search, we can utilize the LLM to select out new top-N reasoning paths P ending with the tail relations R^D from P_{cand} based on the literal information of the question x and the candidate relations R^D_{cand}. The prompt used here can be found in Appendix E.3.1. As shown in Figure 2, the LLM selects top-3 relations $\{$**capital of**, **country**, **territory**$\}$ out from all relations linked to the entity **Canberra** in the first iteration. Since **Canberra** is the only topic entity, the top-3 candidate reasoning paths are updated as $\{$(**Canberra, capital of**), (**Canberra, country**),(**Canberra, territory**)$\}$.

Entity Exploration Similar to relationship exploration, entity exploration is also a beam search process performed by the LLM from R^D to E^D, and consists of two steps, `Search` and `Prune`.

- **Search** Once we have obtained new top-N reasoning paths P and the set of new tail relations R^D from relation exploration, for each relation path $p_n \in P$, we can explore a candidate entity set $E^D_{cand,n}$ by querying $(e_n^{D-1}, r_n^D, ?)$ or $(?, r_n^D, e_n^{D-1})$, where e_n^{D-1}, r_n denote the tail entity and relation of p_n. We can aggregate $\{E^D_{cand,1}, E^D_{cand,2}, ..., E^D_{cand,N}\}$ into E^D_{cand} and expand top-N reasoning paths P to P_{cand} with the tail entities E^D_{cand}. For the shown case, E^1_{cand} can be represented as $\{$**Australia**, **Australia**, **Australian Capital Territory**$\}$.
- **Prune** Since the entities in each candidate set E^D_{cand} is expressed in natural language, we can leverage the LLM to select new top-N reasoning paths P ending with the tail entities E^D out from P_{cand}. The prompt used here can be found in Appendix E.3.2. As shown in Figure 2, **Australia** and **Australian Capital Territory** are scored as 1 since the relations **capital of**, **country** and **territory** are only linked to one tail entity respectively, and the current reasoning paths p are updated as $\{$(**Canberra, capital of, Australia**), (**Canberra, country, Australia**), (**Canberra, territory, Australian Capital Territory**)$\}$.

After executing the two explorations described above, we reconstruct new top-N reasoning paths P where the length of each path increases by 1. Each prune step requires at most N LLM calls.

2.1.3 REASONING

Upon obtaining the current reasoning path P through the exploration process, we prompt the LLM to evaluate whether the current reasoning paths are adequate for generating the answer. If the evaluation yields a positive result, we prompt the LLM to generate the answer using the reasoning paths with the query as inputs as illustrated in Figure 2. The prompt used for evaluation and generation can be found in Appendix E.3.3 and E.3.4. Conversely, if the evaluation yields a negative result, we repeat the `Exploration` and `Reasoning` steps until the evaluation is positive or reaches the maximum

- **搜索** 一旦我们从关系探索 (relation exploration) 中获得新的前 N 推理路径 P 和新的尾关系集合 R^D，对于每个关系路径 $p_n \in P$，我们可以通过查询 $(e_n^{D-1}, r_n^D, ?)$ 或 $(?, r_n^D, e_n^{D-1})$ 来探索候选实体集合 $E^D_{cand,n}$，其中 e_n^{D-1}, r_n 表示 p_n 的尾实体和关系。我们可以将 $\{E^D_{cand,1}, E^D_{cand,2}, ..., E^D_{cand,N}\}$ 聚合成 E^D_{cand}，并将前 N 推理路径 P 扩展到 P_{cand}，使用尾实体 E^D_{cand}。对于所示的情况，E^1_{cand} 可以表示为 $\{$**Australia**, **Australia**, **Australian Capital Territory**$\}$。

- **修剪** 由于每个候选集 E^D_{cand} 中的实体以自然语言表达，我们可以利用大语言模型 (LLM) 从 P_{cand} 中选择新的前 N 个推理路径 P，这些路径以尾实体 E^D 结束。这里使用的提示可以在附录 E.3.2 中找到。如图 2 所示，**澳大利亚** 和 **澳大利亚首都领地** 的得分为 1，因为关系 **首都**、**国家** 和 **领地** 仅分别与一个尾实体相连，而当前的推理路径 p 更新为 $\{$(**堪培拉, 首都, 澳大利亚**), (**堪培拉, 国家, 澳大利亚**), (**堪培拉, 领地, 澳大利亚首都领地**)$\}$.

执行上述两个探索（exploration）后，我们重构新的前N个推理路径P，其中每条路径的长度增加1。每个剪枝步骤最多需要N次LLM调用。

2.1.3 REASONING

在通过探索（exploration）过程获得当前推理路径 P 后，我们提示 LLM 评估当前推理路径是否足以生成答案。如果评估结果为积极，我们提示 LLM 使用推理路径和查询作为输入生成答案，如图 2 所示。用于评估和生成的提示可以在附录 E.3.3 和 E.3.4 中找到。相反，如果评估结果为消极，我们将重复 `Exploration` 和 `Reasoning` 步骤，直到评估结果为积极或达到最大搜索深度 D_{max}。如果算法尚未结束，则表示即便达到 D_{max}，ToG 仍无法探索推理路径以解决问题。在这种情况下，ToG 仅基于 LLM 中的固有知识生成答案。ToG 的整个推理过程包含 D 次探索（exploration）阶段和 D 次评估步骤，以及一次生成步骤，最多需要 $2ND + D + 1$ 次调用 LLM。

2.2 RELATION-BASED THINK-ON-GRAPH

以前的 KBQA 方法，尤其是基于语义解析的方法，主要依赖于问题中的关系信息来生成正式查询 (Lan et al., 2022)。受到此启发，我们提出了基于关系的 ToG (ToG-R)，该方法探索以主题实体 $\{e_n^0\}_{n=1}^N$ 开始的前 N 个关系链 $\{p_n = (e_n^0, r_n^1, r_n^2, ..., r_n^D)\}_{n=1}^N$，而不是基于三元组的推理路径。ToG-R 在每次迭代中顺序执行 关系搜索、实体剪枝和 实体搜索，这与 ToG 相同。然后，ToG-R 基于实体搜索获得的以 E^D_{cand} 结束的所有候选推理路径执行 推理步骤。如果 LLM 判断检索到的候选推理路径没有包含足够的信息供 LLM 回答问题，我们将从候选实体 E^D_{cand} 随机抽样 N 个实体，并继续进行下一次迭代。假设每个实体集 $E^D_{cand,n}$ 中的实体可能属于同一实体类别，并具有相似的邻接关系，则修剪实体集 $\{E^D_{cand,n}\}_{n=1}^N$ 的结果可能对后续的关系探索影响不大。因此，我们在 实体剪枝 中使用随机束搜索，而不是 ToG 中的 LLM 约束束搜索，称为 **随机剪枝**。算法 1 和 2 显示了 ToG 和 ToG-R 的实现细节。ToG-R 最多需要 $ND + D + 1$ 次调用 LLM。

与 ToG 相比，ToG-R 提供了两个关键利益：1) 它消除了使用 LLM 剪枝实体的过程，从而减少了整体成本和推理时间。2) ToG-R 主要强调关系的字面信息，减轻了在中间实体的字面信息缺失或 LLM 不熟悉时导致错误推理的风险。

search depth D_{max}. If the algorithm has not yet concluded, it signifies that even upon reaching the D_{max}, ToG remains unable to explore the reasoning paths to resolve the question. In such a scenario, ToG generates the answer exclusively based on the inherent knowledge in the LLM. The whole inference process of ToG contains D exploration phases and D evaluation steps as well as a generation step, which needs at most $2ND + D + 1$ calls to the LLM.

2.2 Relation-based Think-on-Graph

Previous KBQA methods, particularly based on semantic parsing, have predominantly relied on relation information in questions to generate formal queries (Lan et al., 2022). Inspired by this, we propose relation-based ToG (ToG-R) that explores the top-N relation chains $\{p_n = (e_n^0, r_n^1, r_n^2, ..., r_n^D)\}_{n=1}^N$ starting with the topic entities $\{e_n^0\}_{n=1}^N$ instead of triple-based reasoning paths. ToG-R sequentially performs relation search, relation prune and entity search in each iteration, which is the same as ToG. Then ToG-R performs the reasoning step based on all candidate reasoning paths ending with E_{cand}^D obtained by entity search. If the LLM determines that the retrieved candidate reasoning paths do not contain enough information for the LLM to answer the question, we randomly sample N entities from the candidate entities E_{cand}^D and continue to the next iteration. Assuming that entities in each entity set $E_{cand,n}^D$ probably belong to the same entity class and have similar neighboring relations, the results of pruning the entity set $\{E_{cand,n}^D\}_{n=1}^N$ might have little impact on the following relation exploration. Thus, we use the random beam search instead of the LLM-constrained beam search in ToG for entity prune, referred to as **random prune**. Algorithm 1 and 2 show the implementation details of the ToG and ToG-R. ToG-R needs at most $ND + D + 1$ calls to the LLM.

Compared to ToG, ToG-R offers two key benefits: 1) It eliminates the need for the process of pruning entities using the LLM, thereby reducing the overall cost and reasoning time. 2) ToG-R primarily emphasizes the literal information of relations, mitigating the risk of misguided reasoning when the literal information of intermediate entities is missing or unfamiliar to the LLM.

3 Experiments

3.1 Experimental Design

3.1.1 Datasets and Evaluation Metrics

In order to test ToG's ability on multi-hop knowledge-intensive reasoning tasks, we evaluate ToG on five KBQA datasets (4 Multi-hop and 1 Single-hop): CWQ (Talmor & Berant, 2018), WebQSP (Yih et al., 2016), GrailQA (Gu et al., 2021), QALD10-en (Perevalov et al., 2022), Simple Questions (Bordes et al., 2015). Moreover, in order to examine ToG on more generic tasks, we also prepare one open-domain QA dataset: WebQuestions (Berant et al., 2013); two slot filling datasets: T-REx (ElSahar et al., 2018) and Zero-Shot RE (Petroni et al., 2021); and one fact-checking dataset: Creak (Onoe et al., 2021). Note that, for two big datasets GrailQA and Simple Questions, we only randomly selected 1,000 samples each for testing in order to save computational cost. For all datasets, exact match accuracy (Hits@1) is used as our evaluation metric following previous works (Li et al., 2023c; Baek et al., 2023b; Jiang et al., 2023; Li et al., 2023a).

Method	Multi-Hop KBQA				Single-Hop KBQA	Open-Domain QA	Slot Filling		Fact Checking
	CWQ	WebQSP	GrailQA	QALD10-en	Simple Questions	WebQuestions	T-REx	Zero-Shot RE	Creak
Without external knowledge									
IO prompt w/ChatGPT	37.6	63.3	29.4	42.0	20.0	48.7	33.6	27.7	89.7
CoT w/ChatGPT	38.8	62.2	28.1	42.9	20.3	48.5	32.0	28.8	90.1
SC w/ChatGPT	45.4	61.1	29.6	45.3	18.9	50.3	41.8	45.4	90.8
With external knowledge									
Prior FT SOTA	70.4[α]	82.1[β]	75.4[γ]	45.4[δ]	85.8[ϵ]	56.3[ζ]	87.7[η]	74.6[θ]	88.2[ι]
Prior Prompting SOTA	-	74.4[κ]	53.2[κ]	-	-	-	-	-	-
ToG-R (Ours) w/ChatGPT	58.9	75.8	56.4	48.6	45.4	53.2	75.3	86.5	93.8
ToG (Ours) w/ChatGPT	57.1	76.2	68.7	50.2	53.6	54.5	76.8	88.0	91.2
ToG-R (Ours) w/GPT-4	**69.5**	81.9	80.3	**54.7**	58.6	57.1	75.5	86.9	95.4
ToG (Ours) w/GPT-4	67.6	**82.6**	**81.4**	53.8	**66.7**	**57.9**	**77.1**	**88.3**	**95.6**

表 1: 不同数据集的 ToG 结果。先前的 FT（微调）和提示 SOTA 包括已知的最佳结果：α: Das et al. (2021); β: Yu et al. (2023); γ: Gu et al. (2023); δ: Santana et al. (2022); ϵ: Baek et al. (2023a); ζ: Kedia et al. (2022); η: Glass et al. (2022); θ: Petroni et al. (2021); ι: Yu et al. (2022); κ: Li et al. (2023a).

3 Experiments

3.1 Experimental Design

3.1.1 Datasets and Evaluation Metrics

为了测试ToG在多跳知识密集推理任务上的能力，我们在五个KBQA数据集（4个多跳和1个单跳）上评估ToG：CWQ (Talmor & Berant, 2018)，WebQSP (Yih et al., 2016)，GrailQA (Gu et al., 2021)，QALD10-en (Perevalov et al., 2022)，Simple Questions (Bordes et al., 2015)。此外，为了检验ToG在更通用任务上的表现，我们还准备了一个开放领域问答数据集：WebQuestions (Berant et al., 2013)；两个槽填充数据集：T-REx (ElSahar et al., 2018)和零样本RE（Zero-Shot RE）(Petroni et al., 2021)；以及一个事实核查数据集：Creak (Onoe et al., 2021)。请注意，对于两个大数据集GrailQA和Simple Questions，我们仅随机选择了各1,000个样本进行测试，以节省计算成本。对于所有数据集，采用精确匹配准确率（Hits@1）作为我们的评估指标，遵循之前的研究 (Li et al., 2023c; Baek et al., 2023b; Jiang et al., 2023; Li et al., 2023a)。

3.1.2 Methods Selected for Comparison

我们与标准提示（IO prompt）(Brown et al., 2020b)、思维链提示（CoT prompt）(Wei et al., 2022) 和自一致性（Self-Consistency）(Wang et al., 2023c) 进行比较，使用 6 个上下文示例和“逐步”推理链。此外，对于每个数据集，我们选择之前的状态-of-the-art（SOTA）作品进行比较。我们注意到，专门在评估数据集上训练的微调方法本质上通常具有相对于未训练的基于提示的方法的优势，但在其他数据上牺牲了灵活性和泛化能力。因此，为了公平起见，我们分别与所有基于提示的方法中的之前 SOTA 和所有方法中的之前 SOTA 进行比较。请注意，论文 Tan et al. (2023) 不参与比较，因为其结果不基于标准的精确匹配，因此无法比较。

3.1.3 Experiment Details

Method	Multi-Hop KBQA				Single-Hop KBQA	Open-Domain QA	Slot Filling		Fact Checking
	CWQ	WebQSP	GrailQA	QALD10-en	Simple Questions	WebQuestions	T-REx	Zero-Shot RE	Creak
Without external knowledge									
IO prompt w/ChatGPT	37.6	63.3	29.4	42.0	20.0	48.7	33.6	27.7	89.7
CoT w/ChatGPT	38.8	62.2	28.1	42.9	20.3	48.5	32.0	28.8	90.1
SC w/ChatGPT	45.4	61.1	29.6	45.3	18.9	50.3	41.8	45.4	90.8
With external knowledge									
Prior FT SOTA	70.4[α]	82.1[β]	75.4[γ]	45.4[δ]	85.8[ϵ]	56.3[ζ]	87.7[η]	74.6[θ]	88.2[ι]
Prior Prompting SOTA	-	74.4[κ]	53.2[κ]	-	-	-	-	-	-
ToG-R (Ours) w/ChatGPT	58.9	75.8	56.4	48.6	45.4	53.2	75.3	86.5	93.8
ToG (Ours) w/ChatGPT	57.1	76.2	68.7	50.2	53.6	54.5	76.8	88.0	91.2
ToG-R (Ours) w/GPT-4	**69.5**	81.9	80.3	**54.7**	58.6	57.1	75.5	86.9	95.4
ToG (Ours) w/GPT-4	67.6	**82.6**	**81.4**	53.8	**66.7**	**57.9**	**77.1**	**88.3**	**95.6**

表 1: The ToG results for different datasets. The prior FT (Fine-tuned) and prompting SOTA include the best-known results: α: Das et al. (2021); β: Yu et al. (2023); γ: Gu et al. (2023); δ: Santana et al. (2022); ϵ: Baek et al. (2023a); ζ: Kedia et al. (2022); η: Glass et al. (2022); θ: Petroni et al. (2021); ι: Yu et al. (2022); κ: Li et al. (2023a).

3.1.2 Methods Selected for Comparison

We compare with standard prompting (IO prompt) (Brown et al., 2020b), Chain-of-Thought prompting (CoT prompt) (Wei et al., 2022), and Self-Consistency (Wang et al., 2023c) with 6 in-context exemplars and "step-by-step" reasoning chains. Moreover, for each dataset, we pick previous state-of-the-art (SOTA) works for comparison. We notice that fine-tuning methods trained specifically on evaluated datasets usually have an advantage by nature over methods based on prompting without training, but sacrificing the flexibility and generalization on other data. For a fair play, therefore, we compare with previous SOTA among all prompting-based methods and previous SOTA among all methods respectively. Note that the paper Tan et al. (2023) is not involved in comparison because its results are not based on standard exact match and thus incomparable.

3.1.3 Experiment Details

Given the plug-and-play convenience of ToG, we try three LLMs in experiments: ChatGPT, GPT-4 and Llama-2. We use OpenAI API to call ChatGPT (GPT-3.5-turbo) and GPT-4[1]. Llama-2-70B-Chat (Touvron et al., 2023) runs with 8 A100-40G without quantization, where the temperature parameter is set to 0.4 for exploration process (increasing diversity) and set to 0 for reasoning process (guaranteeing reproducibility). The maximum token length for the generation is set to 256. In all experiments, we set both width N and depth D_{max} to 3 for beam search. Freebase (Bollacker et al., 2008) is used as KG

Method	**CWQ**	**WebQSP**
Fine-tuned		
NSM (He et al., 2021)	53.9	74.3
CBR-KBQA (Das et al., 2021)	67.1	-
TIARA (Shu et al., 2022)	-	75.2
DeCAF (Yu et al., 2023)	70.4	82.1
Prompting		
KD-CoT (Wang et al., 2023b)	50.5	73.7
StructGPT (Jiang et al., 2023)	-	72.6
KB-BINDER (Li et al., 2023a)	-	74.4
LLama2-70B-Chat		
CoT	39.1	57.4
ToG-R	**57.6**	**68.9**
ToG	53.6	63.7
Gain	(+18.5)	(+11.5)
ChatGPT		
CoT	38.8	62.2
ToG-R	57.1	75.8
ToG	**58.9**	**76.2**
Gain	(+20.1)	(+14.0)
GPT-4		
CoT	46.0	67.3
ToG-R	67.6	81.9
ToG	**69.5**	**82.6**
Gain	(+23.5)	(+15.3)

[1]GPT-3.5-turbo and GPT-4 is both from https://openai.com/

鉴于ToG的即插即用便利性，我们在实验中尝试了三种大语言模型（LLMs）：ChatGPT、GPT-4和Llama-2。我们使用OpenAI API调用ChatGPT（GPT-3.5-turbo）和GPT-4[1]。Llama-2-70B-Chat (Touvron et al., 2023)在无量化条件下使用8个A100-40G运行，其中温度参数在探索过程（增加多样性）中设置为0.4，在推理过程（保证可重复性）中设置为0。生成的最大标记（Token）长度设置为256。在所有实验中，我们将宽度N和深度D_{max}均设置为3，以进行束搜索。Freebase (Bollacker et al., 2008)作为CWQ、WebQSP、GrailQA、Simple Questions和Webquestions的知识图谱（KG），而Wikidata (Vrandečić & Krötzsch, 2014)作为QALD10-en、T-REx、零样本（Zero-Shot）关系提取（RE）和Creak的知识图谱（KG）。我们在所有数据集的ToG推理提示中使用5个样本（shots）。

3.2 Main Results

3.2.1 Comparison to Other Methods

表 2: 在CWQ和WebQSP上使用不同骨干模型 (backbone models) 的ToG (ToG) 的性能 (performances)。

由于 CoT 使用外部知识图谱 (KG) 来增强大型语言模型 (LLM)，我们首先将其与那些同样利用外部知识的方法进行比较。如图 1 所示，即使 ToG 是一种无训练的基于提示的方法，并且在与那些经过数据训练来评估的微调方法的比较中存在天然劣势，ToG 与 GPT-4 仍在 9 个数据集中有 6 个达到了新的最先进技术 (SOTA) 性能，包括 WebQSP、GrailQA、QALD10-en、WebQuestions、零样本 (Zero-Shot) RE 和 Creak。即使是在一些没有 SOTA 的数据集上，例如 CWQ，CoT 的性能也已接近 SOTA (69.5% 对 70.4%)。如果与所有基于提示的方法进行比较，ToG 与 GPT-4 及其较弱版本 ToG 与 ChatGPT 在所有数据集中均能赢得竞争。尤其是在开放域问答数据集 WebQuestions 上，1.6% 的提升证明了 ToG 在开放域问答任务上的通用性。我们还注意到，ToG 在单跳知识库问答 (KBQA) 数据集上的表现不及其在其他数据集上的表现。这些结果表明，ToG 在多跳数据集上的有效性普遍较高，这支持了我们关于 ToG 增强 LLM 深层推理能力的论点。

我们也从图 1 中看到，与那些不利用外部知识的方法（例如 IO、CoT 和 SC 提示方法）相比，ToG 的优势更加显著。例如，GrailQA 和零样本 (Zero-Shot) RE 的性能分别提高了 51.8% 和 42.9%。事实证明，外部 KG 在推理中的好处是不容忽视的。

ToG 在大多数数据集上优于 ToG-R，因为与 ToG-R 提取的关系链相比，基于三元组的推理路径提供了额外的中间实体信息。有关 ToG 生成的答案的更详细分析可以在附录 B.2 中查看。为了更好地比较，之前方法在每个数据集上的结果在附录 C 中进行了报告。

[1]GPT-3.5-turbo和GPT-4均来自https://openai.com/

for CWQ, WebQSP, GrailQA, Simple Questions, and Webquestions, and Wikidata (Vrandečić & Krötzsch, 2014) is used as KG for QALD10-en, T-REx, Zero-Shot RE and Creak. We use 5 shots in ToG-reasoning prompts for all the datasets.

3.2 Main Results

3.2.1 Comparison to Other Methods

Since CoT uses external KG to enhance LLM, we first compare it with those methods leveraging external knowledge as well. As we can see in Figure 1, even if ToG is a training-free prompting-based method and has natural disadvantage in comparison with those fine-tuning methods trained with data for evaluation, ToG with GPT-4 still achieves new SOTA performance in 6 out of 9 datasets, including WebQSP, GrailQA, QALD10-en, WebQuestions, Zero-Shot RE and Creak. Even for some dataset without SOTA, e.g., CWQ, the performance of CoT has already been close to SOTA (69.5% v.s. 70.4%). If comparing with all promoting-based methods, both ToG with GPT-4 and its weaker version ToG with ChatGPT can win the competition in all datasets. In particular, the improvement of 1.6% on open-domain QA dataset WebQuestions demonstrates the ToG's generality on open-domain QA tasks. We also notice that the performance of ToG on single-hop KBQA dataset is not as good as its performance on other datasets. These results indicate that ToG is more effective on multi-hop datasets in general, which supports our argument that ToG enhances the deep reasoning capability of LLMs.

We also see from Figure 1 that, compared with those methods without leveraging external knowledge (e.g, IO, CoT and SC prompting methods), the advantage of ToG is more significant. For example, the performance improves 51.8% and 42.9% on GrailQA and Zero-Shot RE, respectively. It turns out that benefits from external KG can not be ignored in reasoning.

ToG outperforms ToG-R on most datasets since the triple-based reasoning paths provide additional intermediate entity information compared to the relation chains retrieved by ToG-R. More detailed analysis of the answers generated by ToG can be checked in Appendix B.2. And the results of previous methods on each dataset are reported in Appendix C for better comparison,

3.2.2 Performances with Different Backbone Models

Given ToG's flexibility of plug-and-play, we evaluate how different backbone models affect its performance on two datasets CWQ and WebQSP. Table 2 shows that, as we expected, the performance of CoT improves with the size (also reflecting partially the reasoning ability) of backbone models (GPT-4 > ChatGPT > Llama-2). Furthermore, we see that, the larger the backbone model, the larger the gap between CoT and ToG (the gain increases from 18.5% for Llama-2 to 23.5% for GPT-4 on CWQ, and from 11.5% for Llama-2 to 15.3% for GPT-4 on WebQSP), and this indicates more potential of KG can be mined using a more powerful LLM.

3.2.2 Performances with Different Backbone Models

鉴于ToG的即插即用灵活性，我们评估了不同骨干模型对其在两个数据集CWQ和WebQSP上的性能影响。表 2 显示，正如我们预期的那样，CoT的性能随着骨干模型的规模（也在一定程度上反映了推理能力）（模型(model)：GPT-4 > ChatGPT > Llama-2）而提升。此外，我们发现，骨干模型越大，CoT与ToG之间的差距越大（在CWQ上，Llama-2的增益从18.5%增加到GPT-4的23.5%，而在WebQSP上，从Llama-2的11.5%增加到GPT-4的15.3%），这表明使用更强大的LLM可以挖掘出更多KG的潜力。

此外，即便使用最小的模型Llama-2（70B参数），ToG的性能仍然优于使用GPT-4的CoT。这意味着LLM部署和应用的一种更便宜的技术路线，即使用便宜的小型LLM的ToG可能成为替代昂贵大型LLM的候选方案，特别是在外部KG能够覆盖的垂直场景中。

3.2.3 Ablation Study

我们进行各种消融研究（ablation studies）以理解不同因素在ToG（任务导向生成）中的重要性。我们在CWQ和WebQSP的测试集的两个子集上进行消融研究，每个子集包含1000个随机抽样的问题。

搜索深度和宽度对ToG重要吗？ 为探索搜索深度 D_{max} 和波束宽度 N 对 ToG 性能的影响，我们在深度范围从 1 到 4 及宽度从 1 到 4 的设置下进行实验。如图 3 所示，ToG 的性能随着搜索深度和宽度的增加而提高。这也意味着 ToG 的性能可能会随着探索深度和广度的增加而得到提升。然而，考虑到计算成本（随着深度线性增加），我们将深度和宽度均设置为 3，作为默认实验设置。另一方面，当深度超过 3 时，性能增长趋于减弱。这主要是因为只有一小部分问题的推理深度（基于 SPARQL 中的关系数量，如附录中的图 12 所示）大于 3。

Method	CWQ	WebQSP
CoT	37.6	62.0
ToG		
w/ Freebase	58.8	76.2
w/ WikiData	54.9	68.6
ToG-R		
w/ Freebase	59.2	75.1
w/ WikiData	51.9	66.7

表 3: 使用不同源知识图谱（KGs）在CWQ和WebQSP上进行的ToG性能。

不同的知识图谱 (KG) 是否影响 ToG 的性能？ ToG 的主要优势之一是其即插即用的能力。如表 3 所示，与 CoT 相比，ToG 在 CWQ 和 WebQSP 上与不同源 KG 取得了显著的提升。另一方面，不同源 KG 对 ToG 的性能可能会产生不同的影响。值得注意的是，Freebase 在 CWQ 和 WebQSP 上带来的改善比 Wikidata 更为显著，因为这两个数据集都是基于 Freebase 构建的。此外，在像 Wikidata 这样的大型 KG 中，搜索和修剪过程相对具有挑战性。

不同的提示设计如何影响 ToG？ 我们进行额外实验，以确定哪些类型的提示表征（representation）可以与我们的方法良好配合。结果如表 4所示。“三元组”表示使用三元组格式作为提示以表征多个路径，例如“(Canberra, capital of, Australia), (Australia, prime minister, Anthony Albanese)”。“序列”指的是利用序列格式，如图 2所示。“句子”涉及将三

Method	CWQ	WebQSP
ToG		
w/ Triples	58.8	76.2
w/ Sequences	57.2	73.2
w/ Sentences	58.6	73
ToG-R		
w/ Sequences	59.2	75.1
w/ Sentences	50.1	67.3

In addition, even if using the smallest model Llama-2 (70B parameters), ToG outperforms CoT with GPT-4. This implies a much cheaper technical route for LLM deployment and application, i.e., TOG with cheap small LLM may be a candidate for substituting expensive big LLM, especially in vertical scenarios that external KGs can cover.

3.2.3 Ablation Study

We perform various ablation studies to understand the importance of different factors in ToG. We conduct our ablation studies on two subsets of the test sets of CWQ and WebQSP, each of which contains 1,000 randomly sampled questions.

Do search depth and width matter for ToG? To explore the influence of the search depth D_{max} and the beam width N on ToG's performance, we conduct experiments under settings with depths ranging from 1 to 4 and widths from 1 to 4. As shown in Figure 3, ToG's performance improves with the search depth and width. This also implies that ToG's performance could potentially be improved with the increment of the exploration depth and breadth. However, considering the computational cost (which increases linearly with the depth), we set both the depth and width to 3 as the default experimental setting. On the other hand, the performance growth diminishes when the depth exceeds 3. This is mainly because only a small part of questions have the reasoning depths (based on the number of relations in SPARQL, as seen in Figure 12 in the Appendix) of greater than 3.

Method	CWQ	WebQSP
CoT	37.6	62.0
ToG		
w/ Freebase	58.8	76.2
w/ WikiData	54.9	68.6
ToG-R		
w/ Freebase	59.2	75.1
w/ WikiData	51.9	66.7

表 3: Performances of ToG using different source KGs on CWQ and WebQSP.

Do different KGs affect ToG's performance? One of the main advantages of ToG is its plug-and-play capabilities. As shown in Table 3, ToG achieves significant improvements with different source KGs on CWQ and WebQSP, compared to CoT. On the other hand, different source KGs might have different effects on the performance of ToG. Notably, Freebase brings more significant improvements on CWQ and WebQSP than Wikidata, since both datasets are constructed upon Freebase. Moreover, in a very large KG like Wikidata, the searching and pruning processes are relatively challenging.

How do different prompt designs affect ToG? We perform additional experiments to determine which types of prompt representations can work well for our approach. The results are presented in Table 4. "Triples" denotes using triple formats as prompts to represent multiple paths, such as "(Canberra, capital of, Australia), (Australia, prime minister, Anthony Albanese)". "Sequences" refers to the utilization of a sequence format, as illustrated in Figure 2. "Sentences" involves converting the triples into natural language sentences. For example, "(Canberra, capital of, Australia)" can be converted to "The capital of Canberra is Australia." The result shows that the utilization of triple-

Method	CWQ	WebQSP
ToG		
w/ Triples	58.8	76.2
w/ Sequences	57.2	73.2
w/ Sentences	58.6	73
ToG-R		
w/ Sequences	59.2	75.1
w/ Sentences	50.1	67.3

表 4: Performances of ToG using different prompting designs.

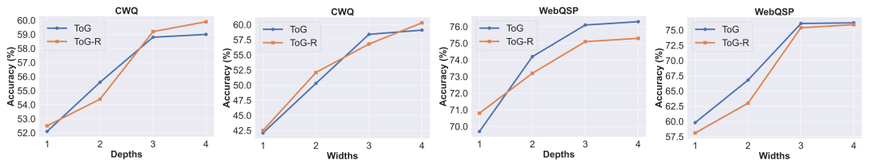

图 3: 不同搜索深度和宽度下的 ToG (Tree of Games) 性能表现。

元组转换为自然语言句子。例如，“(Canberra, capital of, Australia)”可以转换为“The capital of Canberra is Australia.” 结果显示，基于三元组的表征（representation）用于推理路径时，效率最高且表现优越。相反，在考虑ToG-R时，每个推理路径是从主题实体开始的关系链，因此与基于三元组的提示表征（representation）不兼容。因此，将ToG-R转换为自然语言形式会导致提示过长，从而显著降低性能。

比较不同剪枝工具的影响。 其他除了LLM之外，可以在探索阶段作为剪枝工具使用的轻量级模型，如BM25和SentenceBERT，可以根据它们与问题的字面相似度选择前N个实体和关系。我们调查了不同剪枝工具对ToG性能的影响，如表5所示。用BM25或SentenceBERT替代LLM会导致我们的方法性能显著下降。具体而言，CWQ上的结果平均下降8.4%，WebQSP上的结果平均下降15.1%。结果表明，LLM在有效性方面表现最佳，作为剪枝工具。另一方面，使用BM25或SentenceBERT后，我们只需要对LLM进行$D+1$次调用，而不是$2ND+D+1$，正如我们在第2.1.3节中讨论的，这提高了ToG的效率。

Method	CWQ	WebQSP
ToG		
w/BM25	51.4	58.7
w/SentenceBERT	51.7	66.3
w/ChatGPT	58.8	76.2
ToG-R		
w/BM25	49.4	57.3
w/SentenceBERT	50.1	60.1
w/ChatGPT	59.2	75.1

表 5: 使用不同剪枝工具的ToG（Tree of Graphs）表现。

我们还进行了额外的消融研究，探讨了种子示例数量的影响以及ToG与naive（naive）束搜索在知识图谱（KG）上的差异，详细内容见附录B.1。

3.3 Knowledge Traceability and Correctability in ToG

KG 的质量对 ToG 的正确推理非常重要。ToG 的一个有趣特性是知识可追溯性 (knowledge traceability) 和知识可纠正性 (knowledge correctability)，在 LLM 推理过程中，它提供了一种利用 ToG 本身来提高 KG 质量并降低 KG 构建和纠正成本的方法。如图 4 所示，ToG 的显式推理路径可以展示给用户。如果人类用户/专家或其他 LLM 发现 ToG 答案中的潜在错误或不确定性，ToG 有能力追溯并检查推理路径，找到有错误的可疑三元组并进行纠正。

以图 4 中的案例为例。给定输入问题 “吉祥物 Phillie Phanatic 的球队春季训练场地是什么？”，ToG 在第一轮输出错误答案 “Bright House Field”。然后 ToG 追溯所有推理路径，定位错误的原因可能来自第二条推理路径 (Phillie Phanatic $\xrightarrow{\text{Team}}$ Philadelphia Phillies $\xrightarrow{\text{Arena Stadium}}$ Bright House Field)，并分析错误来自于过时三元组中的旧名称 “Specturm Field”，即 “Bright House Field” （*Philadelphia Phillies*, *Arena Stadium*, *Bright House Field*）。根据 ToG 的提示，用户可

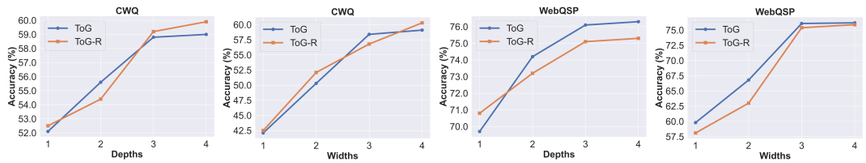

图 3: Performances of ToG with different search depths and widths.

based representations for the reasoning paths yields the highest degree of efficiency and superior performance. Conversely, when considering ToG-R, each reasoning path is a relation chain starting from a topic entity, rendering it incompatible with the triple-based prompt representation. Consequently, the transformation of ToG-R into the natural language form results in excessively lengthy prompts, thereby leading to a notable deterioration in performance.

Comparing the affects from different pruning tools. Other than the LLM, lightweight models that can measure text similarity like BM25 and SentenceBERT, can be employed as pruning tools in the exploration phase. We can select top-N entities and relations based on their literal similarities with the question. We investigate the impacts of different pruning tools on the performance of the ToG, as demonstrated in Table 5. The replacement of the LLM with either BM25 or SentenceBERT results in the significant performance degradation of our approach. Concretely, the results on CWQ drop on average by 8.4%, and the results on WebQSP drop on average by 15.1%. The results show that the LLMs perform best as a pruning tool in terms of effectiveness. On the other hand, after utilizing the BM25 or SentenceBERT, we only need $D + 1$ calls to the LLM instead of $2ND + D + 1$ as we discuss in Section 2.1.3, which enhances the efficiency of ToG.

Method	CWQ	WebQSP
ToG		
w/BM25	51.4	58.7
w/SentenceBERT	51.7	66.3
w/ChatGPT	58.8	76.2
ToG-R		
w/BM25	49.4	57.3
w/SentenceBERT	50.1	60.1
w/ChatGPT	59.2	75.1

表 5: Performances of ToG using different pruning tools.

We conduct additional ablation studies on the effect of the number of seed exemplars and the difference between ToG and naive beam search on the KG, which can be seen in Appendix B.1.

3.3 Knowledge Traceability and Correctability in ToG

The quality of KG is very important for correct reasoning by ToG. An interesting feature of ToG is knowledge traceability and knowledge correctability during LLM reasoning, and it provides a way to improve KG's quality using ToG itself and reduce the cost of KG construction and correction. As illustrated in Figure 4, the explicit reasoning paths of the ToGs can be displayed to users. If potential errors or uncertainties in ToG answers are discovered by human users/experts or other LLMs, ToG has the ability to trace back and examine the reasoning path, find suspicious triples with errors, and correct them.

Take the case in Figure 4 as an example. Given the input question "What is mascot Phillie Phanatic's team's spring training stadium?", ToG outputs the wrong answer "Bright House Field" in the first

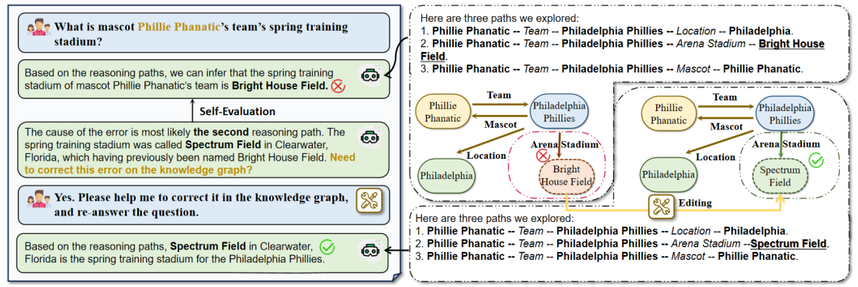

图 4: ToG的知识可追溯性和可修正性的示意图。

以请求 LLM 纠正这个错误，并以正确信息回答同样的问题。这个例子揭示了 ToG 不仅通过 KG 增强了 LLM，还通过 LLM 改善了 KG 的质量，这被称为知识注入 (knowledge infusion) (Moiseev et al., 2022)。

4 Related Work

使用 LLM 提示进行推理 链式思维 (CoT) (Wei et al., 2022) 已被证明在增强 LLM 推理方面是有效的。它根据推理逻辑在少样本学习范式下创建一系列提示实例，以提高 LLM 在复杂任务上的表现。CoT 的思路在不同维度上得到了改善，包括自动链式思维 (Auto-CoT) (Zhang et al., 2022)、复杂链式思维 (Complex-CoT) (Fu et al., 2023)、自一致性 (Self-Consistency) (Wang et al., 2023c)、零样本链式思维 (Zero-Shot-CoT) (Kojima et al., 2022)、迭代链式思维 (Iter-CoT) (Sun et al., 2023b)、思维转移 (ToT) (Yao et al., 2023)、生成链式思维 (GoT) (Besta et al., 2023) 等。考虑到所有这些工作仅使用训练数据中的知识的局限性，最近的努力如 ReAct (Yao et al., 2022) 尝试利用来自外部源的信息，如维基文档，以进一步提升推理性能。

KG增强的 LLM 知识图谱 (KG) 在动态、显式和结构化知识表征 (Pan et al., 2023) 方面具有优势，结合 LLM 和 KG 的技术也得到了研究。早期研究 (Peters et al., 2019; Huang et al., 2024; Luo et al., 2024; Zhang et al., 2021; Li et al., 2023b; Liu et al., 2020) 在预训练或微调过程中将来自 KG 的结构化知识嵌入到底层神经网络中。然而，嵌入 KG 的 LLM 在知识推理的可解释性和知识更新的效率上牺牲了它自身的特性 (Hu et al., 2023)。

最近的工作则将 LLM 与 KG 结合，通过将 KG 中相关的结构化知识翻译成文本提示来增强 LLM 的能力。所有这些方法遵循一个固定的流程，从 KG 中检索额外信息以增强 LLM 提示，并且它们属于我们在引言部分定义的 LLM $\oplus$ KG 范式。另一方面，Jiang et al. (2023) 要求 LLM 探索 KG，因此可以被视为思维转移 (ToG) 的一个特例，属于 LLM $\otimes$ KG 范式。

5 Conclusion

我们引入了 LLM $\otimes$ KG 模式，以紧密耦合的方式集成 LLMs 和 KGs，并提出了 Think-on-Graph (ToG) 算法框架，该框架利用 LLM 作为智能体 (agent) 参与 KG 推理，以实现更好的决策。实验结果表明，ToG 在没有额外训练成本的情况下优于现有的基于微调 (fine-tuning) 和提示 (prompting) 的方法，并且减轻了 LLMs 的幻觉 (hallucination) 问题。

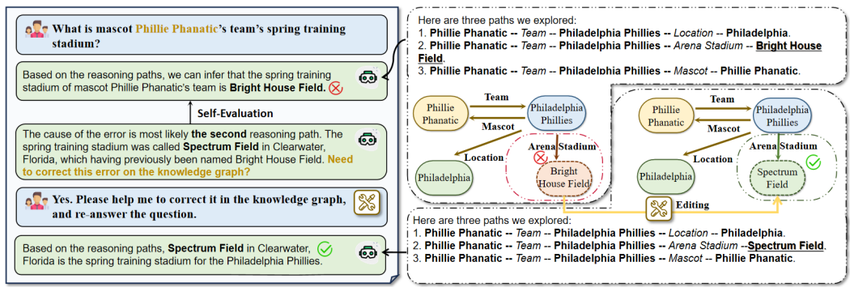

图 4: The illustration of knowledge traceability and correctability of ToG.

round. Then ToG traces back all reasoning paths, localizes the cause of the error may come from the second reasoning path (Phillie Phanatic $\xrightarrow{\text{Team}}$ Philadelphia Phillies $\xrightarrow{\text{Arena Stadium}}$ Bright House Field), and analyzes that the error comes from the old name "Specturm Field" of "Bright House Field" in the outdated triple (*Philadelphia Phillies*, *Arena Stadium*, *Bright House Field*). According to the hints from ToG, user can ask LLM to correct this error and answer the same question with correct information. This example reveals that ToG not only enhances LLM with KG, but also improves the quality of KG with LLM, known as knowledge infusion (Moiseev et al., 2022).

4 Related Work

Reasoning with LLM Prompting Chain-of-Thought (CoT) (Wei et al., 2022) has been shown to be effective in enhancing LLM reasoning. It creates a series of prompt instances according to reasoning logic under a few-shot learning paradigm in order to improve LLM's performance on complex tasks. The thought of CoT has been improved along different dimensions, including Auto-CoT (Zhang et al., 2022), Complex-CoT (Fu et al., 2023), Self-Consistency (Wang et al., 2023c), Zero-Shot-CoT (Kojima et al., 2022), Iter-CoT (Sun et al., 2023b), ToT (Yao et al., 2023), GoT (Besta et al., 2023) and so on. Given the limitation that all these works only use the knowledge in training data, recent efforts such as ReAct (Yao et al., 2022) attempt to utilize the information from external sources such as Wiki documents to further improve the reasoning performance.

KG-enhanced LLM KG has advantages in dynamic, explicit, and structured knowledge representation (Pan et al., 2023) and techniques combining LLMs with KGs have been studied. Early studies (Peters et al., 2019; Huang et al., 2024; Luo et al., 2024; Zhang et al., 2021; Li et al., 2023b; Liu et al., 2020) embed structured knowledge from KGs into the underlying neural networks during the pretraining or fine-tuning process. However, KG embedded in LLM sacrifices its own nature of explainability in knowledge reasoning and efficiency in knowledge updating (Hu et al., 2023).

Recent works instead combine LLMs with KGs by translating relevant structured knowledge from KGs to textual prompts for LLMs. All the methods follow a fixed pipeline that retrieves extra information from KGs to augment the LLM prompt and they belong to the LLM $\oplus$ KG paradigm we defined in the introduction section. On the other hand, Jiang et al. (2023) asks LLM to explore KG and so it can be regarded as a special case of ToG, which belongs to the LLM $\otimes$ KG paradigms.

6 Acknowledgement

我们衷心感谢尊敬的评审专家们对本论文的宝贵反馈和建设性意见，这些对论文的改进和完善有着显著的贡献。他们的深入建议和对细节的认真关注在提升我们研究工作的质量和清晰度方面发挥了关键作用。

参考文献

Farah Atif, Ola El Khatib, and Djellel Eddine Difallah. Beamqa: Multi-hop knowledge graph question answering with sequence-to-sequence prediction and beam search. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (eds.), *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pp. 781–790. ACM, 2023. doi: 10.1145/3539618.3591698. URL https://doi.org/10.1145/3539618.3591698.

Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. Direct fact retrieval from knowledge graphs without entity linking. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 10038–10055. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.acl-long.558. URL https://doi.org/10.18653/v1/2023.acl-long.558.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering, 2023b.

Debayan Banerjee, Pranav Ajit Nair, Ricardo Usbeck, and Chris Biemann. Gett-qa: Graph embedding based t2t transformer for knowledge graph question answering, 2023.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1533–1544. ACL, 2013. URL https://aclanthology.org/D13-1160/.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models, 2023.

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, 2008.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075, 2015. URL http://arxiv.org/abs/1506.02075.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel

5 CONCLUSION

We introduce the LLM ⊗ KG paradigm for integrating LLMs and KGs in a tight-coupling manner, and propose the Think-on-Graph (ToG) algorithmic framework which leverages LLM as a agent participating in KG reasoning for better decision-making. Experimental results demonstrate that ToG outperforms existing fine-tuning-based methods and prompting-based methods without additional training cost and mitigates the hallucination issue of LLMs.

6 ACKNOWLEDGEMENT

We express our sincere gratitude to the esteemed reviewers for their invaluable feedback and constructive comments, which significantly contributed to the improvement and refinement of this paper. Their insightful suggestions and meticulous attention to detail have played a pivotal role in enhancing the quality and clarity of our research work.

参考文献

Farah Atif, Ola El Khatib, and Djellel Eddine Difallah. Beamqa: Multi-hop knowledge graph question answering with sequence-to-sequence prediction and beam search. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (eds.), *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pp. 781–790. ACM, 2023. doi: 10.1145/3539618.3591698. URL https://doi.org/10.1145/3539618.3591698.

Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. Direct fact retrieval from knowledge graphs without entity linking. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 10038–10055. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.acl-long.558. URL https://doi.org/10.18653/v1/2023.acl-long.558.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering, 2023b.

Debayan Banerjee, Pranav Ajit Nair, Ricardo Usbeck, and Chris Biemann. Gett-qa: Graph embedding based t2t transformer for knowledge graph question answering, 2023.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1533–1544. ACL, 2013. URL https://aclanthology.org/D13-1160/.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models, 2023.

M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Shulin Cao, Jiaxin Shi, Zijun Yao, Xin Lv, Jifan Yu, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jinghui Xiao. Program transfer for answering complex questions over knowledge bases. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8128–8140. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.559. URL https://doi.org/10.18653/v1/2022.acl-long.559.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay-Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. Case-based reasoning for natural language queries over knowledge bases, 2021.

Michiel de Jong, Yury Zemlyanskiy, Joshua Ainslie, Nicholas FitzGerald, Sumit Sanghai, Fei Sha, and William Cohen. Fido: Fusion-in-decoder optimized for stronger performance and faster inference. *arXiv preprint arXiv:2212.08153*, 2022.

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, 2008.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075, 2015. URL http://arxiv.org/abs/1506.02075.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Shulin Cao, Jiaxin Shi, Zijun Yao, Xin Lv, Jifan Yu, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jinghui Xiao. Program transfer for answering complex questions over knowledge bases. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8128–8140. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.559. URL https://doi.org/10.18653/v1/2022.acl-long.559.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira,

Cicero Nogueira dos Santos, Zhe Dong, Daniel Cer, John Nham, Siamak Shakeri, Jianmo Ni, and Yun hsuan Sung. Knowledge prompts: Injecting world knowledge into language models through soft prompts, 2022.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/632.html.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=yf1icZHC-l9.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2701–2715, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.194. URL https://aclanthology.org/2022.naacl-main.194.

Yu Gu, Sue Kase, Michelle Vanni, Brian M. Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond I.I.D.: three levels of generalization for question answering on knowledge bases. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (eds.), *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pp. 3477–3488. ACM / IW3C2, 2021. doi: 10.1145/3442381.3449992. URL https://doi.org/10.1145/3442381.3449992.

Yu Gu, Xiang Deng, and Yu Su. Don't generate, discriminate: A proposal for grounding language models to real-world environments, 2023.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (eds.), *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pp. 553–561. ACM, 2021. doi: 10.1145/3437963.3441753. URL https://doi.org/10.1145/3437963.3441753.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Rikui Huang, Wei Wei, Xiaoye Qu, Wenfeng Xie, Xianling Mao, and Dangyang Chen. Joint multi-facts reasoning network for complex temporal question answering over knowledge graph. *arXiv preprint arXiv:2401.02212*, 2024.

Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay-Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. Case-based reasoning for natural language queries over knowledge bases, 2021.

Michiel de Jong, Yury Zemlyanskiy, Joshua Ainslie, Nicholas FitzGerald, Sumit Sanghai, Fei Sha, and William Cohen. Fido: Fusion-in-decoder optimized for stronger performance and faster inference. *arXiv preprint arXiv:2212.08153*, 2022.

Cicero Nogueira dos Santos, Zhe Dong, Daniel Cer, John Nham, Siamak Shakeri, Jianmo Ni, and Yun hsuan Sung. Knowledge prompts: Injecting world knowledge into language models through soft prompts, 2022.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/632.html.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=yf1icZHC-l9.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2701–2715, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.194. URL https://aclanthology.org/2022.naacl-main.194.

Yu Gu, Sue Kase, Michelle Vanni, Brian M. Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond I.I.D.: three levels of generalization for question answering on knowledge bases. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (eds.), *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pp. 3477–3488. ACM / IW3C2, 2021. doi: 10.1145/3442381.3449992. URL https://doi.org/10.1145/3442381.3449992.

Yu Gu, Xiang Deng, and Yu Su. Don't generate, discriminate: A proposal for grounding language models to real-world environments, 2023.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (eds.), *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual*

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data, 2023.

Dan Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009. ISBN 9780135041963. URL https://www.worldcat.org/oclc/315913020.

Akhil Kedia, Mohd Abbas Zaidi, and Haejun Lee. Fie: Building a global probability space by leveraging early fusion in encoder for open-domain question answering. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 4246–4260. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.285. URL https://doi.org/10.18653/v1/2022.emnlp-main.285.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Yunshi Lan and Jing Jiang. Query graph generation for answering multi-hop complex questions from knowledge bases. Association for Computational Linguistics, 2020.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. Few-shot in-context learning on knowledge base question answering. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 6966–6980. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.acl-long.385. URL https://doi.org/10.18653/v1/2023.acl-long.385.

Wendi Li, Wei Wei, Xiaoye Qu, Xian-Ling Mao, Ye Yuan, Wenfeng Xie, and Dangyang Chen. Trea: Tree-structure reasoning schema for conversational recommendation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2970–2982, 2023b.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty, and Soujanya Poria. Chain of knowledge: A framework for grounding large language models with structured knowledge bases, 2023c.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. Reasoning step-by-step: Temporal sentence localization in videos via deep rectification-modulation network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1841–1851, 2020.

Event, Israel, March 8-12, 2021, pp. 553–561. ACM, 2021. doi: 10.1145/3437963.3441753. URL https://doi.org/10.1145/3437963.3441753.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Rikui Huang, Wei Wei, Xiaoye Qu, Wenfeng Xie, Xianling Mao, and Dangyang Chen. Joint multi-facts reasoning network for complex temporal question answering over knowledge graph. *arXiv preprint arXiv:2401.02212*, 2024.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data, 2023.

Dan Jurafsky and James H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009. ISBN 9780135041963. URL https://www.worldcat.org/oclc/315913020.

Akhil Kedia, Mohd Abbas Zaidi, and Haejun Lee. Fie: Building a global probability space by leveraging early fusion in encoder for open-domain question answering. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 4246–4260. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.285. URL https://doi.org/10.18653/v1/2022.emnlp-main.285.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Yunshi Lan and Jing Jiang. Query graph generation for answering multi-hop complex questions from knowledge bases. Association for Computational Linguistics, 2020.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. Few-shot in-context learning on knowledge base question answering. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 6966–6980. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.acl-long.385. URL https://doi.org/10.18653/v1/2023.acl-long.385.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BTKAeLqLMw.

Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. Uni-parser: Unified semantic parser for question answering on knowledge base and database. *arXiv preprint arXiv:2211.05165*, 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*, 2024.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. SKILL: structured knowledge infusion for large language models. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 1581–1588. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.113. URL https://doi.org/10.18653/v1/2022.naacl-main.113.

Yasumasa Onoe, Michael J. Q. Zhang, Eunsol Choi, and Greg Durrett. CREAK: A dataset for commonsense reasoning over entity knowledge. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/5737c6ec2e0716f3d8a7a5c4e0de0d9a-Abstract-round2.html.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv Preprint*, 2022. doi: 10.48550/arXiv.2203.02155. URL https://doi.org/10.48550/arXiv.2203.02155.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*, 2023.

A. Perevalov, D. Diefenbach, R. Usbeck, and A. Both. Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers. In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, pp. 229–234, Los Alamitos, CA, USA, jan 2022. IEEE Computer Society. doi: 10.1109/ICSC52841.2022.00045. URL https://doi.ieeecomputersociety.org/10.1109/ICSC52841.2022.00045.

Wendi Li, Wei Wei, Xiaoye Qu, Xian-Ling Mao, Ye Yuan, Wenfeng Xie, and Dangyang Chen. Trea: Tree-structure reasoning schema for conversational recommendation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2970–2982, 2023b.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty, and Soujanya Poria. Chain of knowledge: A framework for grounding large language models with structured knowledge bases, 2023c.

Daizong Liu, Xiaoye Qu, Jianfeng Dong, and Pan Zhou. Reasoning step-by-step: Temporal sentence localization in videos via deep rectification-modulation network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1841–1851, 2020.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BTKAeLqLMw.

Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. Uni-parser: Unified semantic parser for question answering on knowledge base and database. *arXiv preprint arXiv:2211.05165*, 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*, 2024.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. SKILL: structured knowledge infusion for large language models. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 1581–1588. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.113. URL https://doi.org/10.18653/v1/2022.naacl-main.113.

Yasumasa Onoe, Michael J. Q. Zhang, Eunsol Choi, and Greg Durrett. CREAK: A dataset for commonsense reasoning over entity knowledge. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/5737c6ec2e0716f3d8a7a5c4e0de0d9a-Abstract-round2.html.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike,

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1005. URL https://aclanthology.org/D19-1005.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. Kilt: a benchmark for knowledge intensive language tasks, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Manuel Alejandro Borroto Santana, Bernardo Cuteri, Francesco Ricca, and Vito Barbara. SPARQL-QA enters the QALD challenge. In Xi Yan, Meriem Beloucif, and Ricardo Usbeck (eds.), *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022), Hersonissos, Greece, May 29th, 2022*, volume 3196 of *CEUR Workshop Proceedings*, pp. 25–31. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3196/paper3.pdf.

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. TIARA: Multi-grained retrieval for robust question answering over large knowledge base. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8108–8121, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.555. URL https://aclanthology.org/2022.emnlp-main.555.

Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text, 2019.

Hao Sun, Xiao Liu, Yeyun Gong, Anlei Dong, Jingwen Lu, Yan Zhang, Daxin Jiang, Linjun Yang, Rangan Majumder, and Nan Duan. Beamsearchqa: Large language models are strong zero-shot QA solver. *CoRR*, abs/2305.14766, 2023a. doi: 10.48550/arXiv.2305.14766. URL https://doi.org/10.48550/arXiv.2305.14766.

Jiashuo Sun, Yi Luo, Yeyun Gong, Chen Lin, Yelong Shen, Jian Guo, and Nan Duan. Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models, 2023b.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 641–651. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1059. URL https://doi.org/10.18653/v1/n18-1059.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the*

and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv Preprint*, 2022. doi: 10.48550/arXiv.2203.02155. URL https://doi.org/10.48550/arXiv.2203.02155.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*, 2023.

A. Perevalov, D. Diefenbach, R. Usbeck, and A. Both. Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers. In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, pp. 229–234, Los Alamitos, CA, USA, jan 2022. IEEE Computer Society. doi: 10.1109/ICSC52841.2022.00045. URL https://doi.ieeecomputersociety.org/10.1109/ICSC52841.2022.00045.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1005. URL https://aclanthology.org/D19-1005.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. Kilt: a benchmark for knowledge intensive language tasks, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Manuel Alejandro Borroto Santana, Bernardo Cuteri, Francesco Ricca, and Vito Barbara. SPARQL-QA enters the QALD challenge. In Xi Yan, Meriem Beloucif, and Ricardo Usbeck (eds.), *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022), Hersonissos, Greece, May 29th, 2022*, volume 3196 of *CEUR Workshop Proceedings*, pp. 25–31. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3196/paper3.pdf.

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. TIARA: Multi-grained retrieval for robust question answering over large knowledge base. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8108–8121, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.555. URL https://aclanthology.org/2022.emnlp-main.555.

Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text, 2019.

Hao Sun, Xiao Liu, Yeyun Gong, Anlei Dong, Jingwen Lu, Yan Zhang, Daxin Jiang, Linjun Yang, Rangan Majumder, and Nan Duan. Beamsearchqa: Large language models are strong zero-shot QA solver. *CoRR*, abs/2305.14766, 2023a. doi: 10.48550/arXiv.2305.14766. URL https://doi.org/10.48550/arXiv.2305.14766.

Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 4149–4158, 2019. doi: 10.18653/v1/n19-1421. URL https://doi.org/10.18653/v1/n19-1421.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. Evaluation of chatgpt as a question answering system for answering complex questions. *arXiv preprint arXiv:2303.07992*, 2023.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. Lamda: Language models for dialog applications. *CoRR*, 2022. URL https://arxiv.org/abs/2201.08239.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

Jianing Wang, Qiushi Sun, Nuo Chen, Xiang Li, and Ming Gao. Boosting language models reasoning with chain-of-knowledge prompting, 2023a.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering, 2023b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023c. URL https://openreview.net/pdf?id=1PL1NIMMrw.

Jiashuo Sun, Yi Luo, Yeyun Gong, Chen Lin, Yelong Shen, Jian Guo, and Nan Duan. Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models, 2023b.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 641–651. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1059. URL https://doi.org/10.18653/v1/n18-1059.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019. doi: 10.18653/v1/n19-1421. URL https://doi.org/10.18653/v1/n19-1421.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. Evaluation of chatgpt as a question answering system for answering complex questions. *arXiv preprint arXiv:2303.07992*, 2023.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. Lamda: Language models for dialog applications. *CoRR*, 2022. URL https://arxiv.org/abs/2201.08239.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014. ISSN 0001-0782. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv Preprint*, 2022. URL https://arxiv.org/abs/2201.11903.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 602–631, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.39.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. Decomposition enhances reasoning via self-evaluation guided decoding, 2023.

Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-2033. URL https://doi.org/10.18653/v1/p16-2033.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Wang, Zhiguo Wang, and Bing Xiang. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases, 2023.

Wenhao Yu, Chenguang Zhu, Zhihan Zhang, Shuohang Wang, Zhuosheng Zhang, Yuwei Fang, and Meng Jiang. Retrieval augmentation for commonsense reasoning: A unified approach. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 4364–4377, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.294. URL https://aclanthology.org/2022.emnlp-main.294.

Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. Poolingformer: Long document modeling with pooling attention. In *International Conference on Machine Learning*, pp. 12437–12446. PMLR, 2021.

Hang Zhang, Yeyun Gong, Xingwei He, Dayiheng Liu, Daya Guo, Jiancheng Lv, and Jian Guo. Noisy pair corrector for dense retrieval. *arXiv preprint arXiv:2311.03798*, 2023.

Jianing Wang, Qiushi Sun, Nuo Chen, Xiang Li, and Ming Gao. Boosting language models reasoning with chain-of-knowledge prompting, 2023a.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering, 2023b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023c. URL https://openreview.net/pdf?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv Preprint*, 2022. URL https://arxiv.org/abs/2201.11903.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 602–631, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.39.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. Decomposition enhances reasoning via self-evaluation guided decoding, 2023.

Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-2033. URL https://doi.org/10.18653/v1/p16-2033.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Wang, Zhiguo Wang, and Bing Xiang. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases, 2023.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv Preprint*, 2022. doi: 10.48550/arXiv.2210.03493. URL https://doi.org/10.48550/arXiv.2210.03493.

Wenhao Yu, Chenguang Zhu, Zhihan Zhang, Shuohang Wang, Zhuosheng Zhang, Yuwei Fang, and Meng Jiang. Retrieval augmentation for commonsense reasoning: A unified approach. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 4364–4377, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.294. URL https://aclanthology.org/2022.emnlp-main.294.

Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. Poolingformer: Long document modeling with pooling attention. In *International Conference on Machine Learning*, pp. 12437–12446. PMLR, 2021.

Hang Zhang, Yeyun Gong, Xingwei He, Dayiheng Liu, Daya Guo, Jiancheng Lv, and Jian Guo. Noisy pair corrector for dense retrieval. *arXiv preprint arXiv:2311.03798*, 2023.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv Preprint*, 2022. doi: 10.48550/arXiv.2210.03493. URL https://doi.org/10.48550/arXiv.2210.03493.

A ALGORITHM FOR TOG

我们总结了 ToG 和 ToG-R 的综合算法过程，如算法图 1 和 2 所示。

Algorithm 1 ToG

Require: Input x, LLM π, depth limit D_{max} sample limit N.
Initialize $E^0 \leftarrow$ Extract entities on x, $P \leftarrow []$, $M \leftarrow 0$.
while $D \leq D_{max}$ **do**
  R^D_{cand}, $P_{cand} \leftarrow$ Search(x, E^{D-1}, P)
  R^D, $P \leftarrow$ Prune(π, x, R^D_{cand}, P_{cand})
  E^D_{cand}, $P_{cand} \leftarrow$ Search(x, E^{D-1}, R^D, P)
  E^D, $P \leftarrow$ Prune(π, x, E^D_{cand}, P_{cand})
  if Reasoning(π, x, P) **then**
    Generate(π, x, P)
    break
  end if
  Increment D by 1.
end while
if $D > D_{max}$ **then**
  Generate(π, x)
end if

Algorithm 2 ToG-R

Require: Input x, LLM π, depth limit D_{max} sample limit N.
Initialize $E^0 \leftarrow$ Extract entities on x, $P \leftarrow []$, $M \leftarrow 0$.
while $D \leq D_{max}$ **do**
  R^D_{cand}, $P_{cand} \leftarrow$ Search(x, E^{D-1}, P)
  R^D, $P \leftarrow$ Prune(π, x, R^D_{cand}, P_{cand})
  E^D_{cand}, $P_{cand} \leftarrow$ Search(x, E^{D-1}, R^D, P)
  if Reasoning(π, x, P, E^D_{cand}) **then**
    Generate(π, x, P, E^D_{cand})
    break
  end if
  E^D, $P \leftarrow$ Random_Prune(E^D_{cand}, P_{cand})
  Increment D by 1.
end while
if $D > D_{max}$ **then**
  Generate(π, x)
end if

B ADDITIONAL ABLATION STUDY AND EXPERIMENT ANALYSIS

在本节中，除了第 3.2.3 节，我们还进行更多的消融研究（ablation study）实验，并详细分析 ToG（ToG）的实验结果。

B.1 ADDITIONAL ABLATION STUDY

对种子样本数量的敏感性 为了更好地理解ToG对种子样本数量的敏感性，我们采用了图5所示的敏感性分析。我们进行零样本（zero-shot）实验，并从训练集中选择1-6个样本作为少样本（few-shot）设置。在少样本测试中，我们随机选择$\{1, 2, 3, 4, 6\}$中的M个样本作为示教（demonstration），并重复实验三次。随着示教中的样本数量增加，整体表现一般也有所提升。然而，ToG和ToG-R的性能峰值有所不同（ToG在5-shot时性能最佳，而ToG-R在4-shot时性能最佳）。此外，ToG的零样本性能优于ToG-R。这可以归因于ToG已充分探索路径，确保即使在零样本情况下也能获得良好的表现。相反，ToG-R在路径中省略了实体，但其在示教下的平均性能优于ToG。

与 Naive Beam Search 的区别 ToG 与 beam search 略有不同。ToG 使用前 N 条推理路径作为证据，而 naive beam search 仅选择最可信的路径作为唯一的

Search Algorithm	Dataset	EM
Naive Beam Search	CWQ	30.1
	WebQSP	46.1
TOG-R	CWQ	59.2
	WebQSP	75.1
TOG	CWQ	58.8
	WebQSP	76.2

A ALGORITHM FOR TOG

We summarize the comprehensive algorithmic procedure of ToG and ToG-R, as shown in Figure Algorithm 1 and 2.

Algorithm 1 ToG

Require: Input x, LLM π, depth limit D_{max} sample limit N.
Initialize $E^0 \leftarrow$ Extract entities on x, $P \leftarrow []$, $M \leftarrow 0$.
while $D \leq D_{max}$ **do**
  $R^D_{cand}, P_{cand} \leftarrow$ Search(x, E^{D-1}, P)
  $R^D, P \leftarrow$ Prune(π, x, R^D_{cand}, P_{cand})
  $E^D_{cand}, P_{cand} \leftarrow$ Search(x, E^{D-1}, R^D, P)
  $E^D, P \leftarrow$ Prune(π, x, E^D_{cand}, P_{cand})
  if Reasoning(π, x, P) **then**
    Generate(π, x, P)
    break
  end if
  Increment D by 1.
end while
if $D > D_{max}$ **then**
  Generate(π, x)
end if

Algorithm 2 ToG-R

Require: Input x, LLM π, depth limit D_{max} sample limit N.
Initialize $E^0 \leftarrow$ Extract entities on x, $P \leftarrow []$, $M \leftarrow 0$.
while $D \leq D_{max}$ **do**
  $R^D_{cand}, P_{cand} \leftarrow$ Search(x, E^{D-1}, P)
  $R^D, P \leftarrow$ Prune(π, x, R^D_{cand}, P_{cand})
  $E^D_{cand}, P_{cand} \leftarrow$ Search(x, E^{D-1}, R^D, P)
  if Reasoning(π, x, P, E^D_{cand}) **then**
    Generate(π, x, P, E^D_{cand})
    break
  end if
  $E^D, P \leftarrow$ Random_Prune(E^D_{cand}, P_{cand})
  Increment D by 1.
end while
if $D > D_{max}$ **then**
  Generate(π, x)
end if

B ADDITIONAL ABLATION STUDY AND EXPERIMENT ANALYSIS

In this section, we conduct more experiments for ablation study in addition to Section 3.2.3, and analyze experimental results of ToG in detail.

B.1 ADDITIONAL ABLATION STUDY

Sensitivity to the Number of Seed Examplars To better understand how sensitive ToG is sensitivity to the number of seed exemplars, we employ sensitivity analysis shown in Figure 5. We conduct zero-shot experiment and select 1-6 examples from the training set as few-shot setting. In the few-shot tests, we randomly chose M of $\{1, 2, 3, 4, 6\}$ exemplars as demonstrations and replicated the experiments three times. As the number of examples in the demonstrations increases, the overall performance also generally improves. However, the performance peaks for ToG and ToG-R differ (with the best performance for ToG at 5-shot and for ToG-R at 4-shot). Moreover, ToG's zero-shot performance outpaces ToG-R. This can be attributed to ToG having fully completely explored paths, ensuring commendable performance even in zero-shot. In contrast, ToG-R omits entities in the path, but its average performance with demonstrations is superior to ToG.

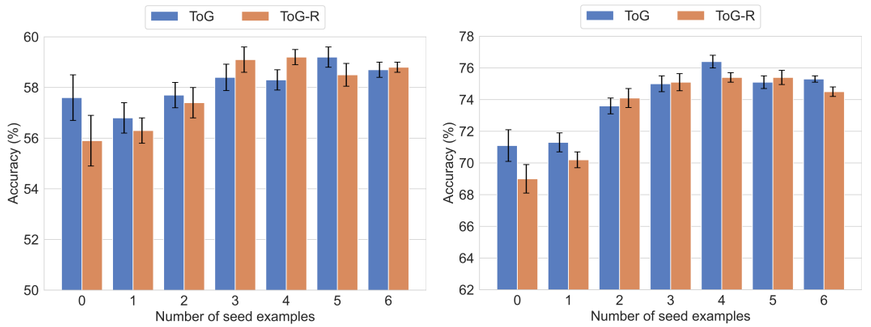

图 5: CWQ 和 WebQSP 对于 ToG 的示例敏感性分析，其中“0” 表示零样本 (zero-shot) 且“k” 表示 k-shot。

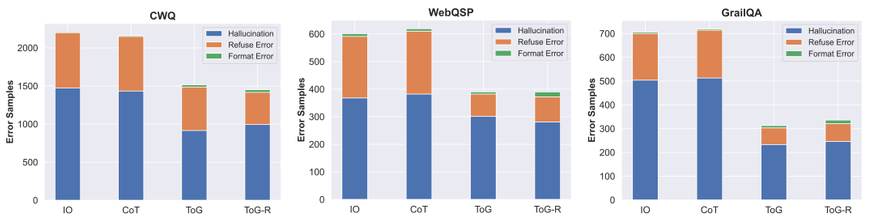

图 6: 在IO、CoT和ToG的CWQ、WebQSP和GrailQA中的错误实例和类别。

推理路径。我们在 CWQ 和 WebQSP 上对 ToG 进行 naive top1-beam search 方法。对于 ToG 的每一层深度，我们选择具有最高可信度的推理路径，以评估当前推理路径是否足以回答问题。实验结果如表 6 所示。在 naive beam search 中，标定误差沿推理累积，从而导致最终结果的不稳定性。我们相信通过考虑前 N 条推理路径，ToG 可以在一定程度上缓解这个问题。

B.2 RESULT ANALYSIS

我们对 ToG 和 ToG-R 生成的答案进行了详细分析。

错误分析 我们考虑了三种类型的错误：（1）幻觉错误（Hallucination error），（2）拒绝错误[2]，和（3）格式错误。这些错误的分布如图 6 所示。我们的方法显著减少了 IO 和 CoT 中的幻觉和拒绝回答错误类型。对于 GrailQA，ToG 甚至将这两种错误类型分别减少了 50% 和 60%。此外，在 ToG 的错误样本中，仍然存在许多幻觉和拒绝回答错误的实例。这是因为当前的搜索深度和宽度都设置为 3。通过增加搜索深度和宽度，这些错误实例将进一步减少（参见 3.2.3 节）。此外，我们目前将不正确的答案概括为幻觉，但幻觉中存在多种类别，我们在本文中不予讨论。此外，在应用 ToG 后，格式错误的样本略有增加。这个结果表明，所探索的路径导致标记（Token）显著增加，有时甚至超过最大输出限制。然而，由于此问题导致的错误率微不足道（少于 3%）。

[2]大型语言模型（LLM）会因信息不足而拒绝回答。

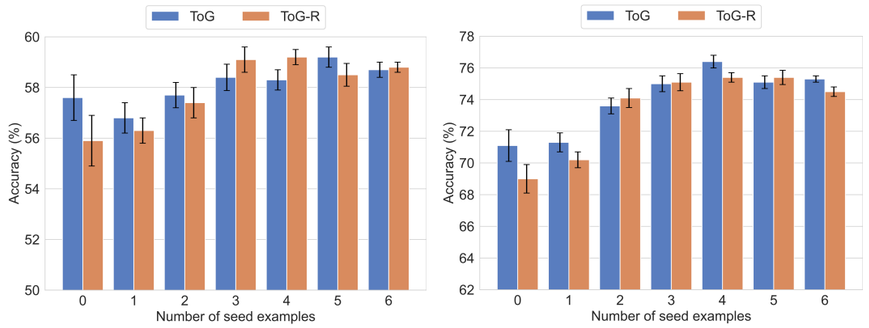

图 5: Exemplar sensitivity analysis for CWQ and WebQSP for ToG, where "0" denotes zero-shot and "k" denotes k-shot.

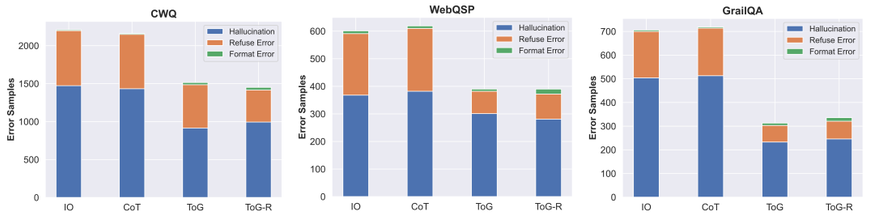

图 6: The erroneous instances and categories in the CWQ, WebQSP, and GrailQA of IO, CoT, and ToG.

Difference with Naive Beam Search ToG is slightly different from the beam search. ToG uses the top-N reasoning paths as evidence while the naive beam search chooses the most plausible path as the only reasoning path. We conduct naive top1-beam search methods for ToG on CWQ and WebQSP. For each depth of the ToG, we choose the reasoning path with the highest plausibility, to evaluate if the current reasoning path is sufficient to answer the questions. The experiment results are shown in Table 6. In naive beam search, the calibration error accumulates along the inference, leading to the instability of the final result. We believe that ToG can partially alleviate this issue by considering the top-N reasoning paths.

Search Algorithm	Dataset	EM
Naive Beam Search	CWQ	30.1
	WebQSP	46.1
TOG-R	CWQ	59.2
	WebQSP	75.1
TOG	CWQ	58.8
	WebQSP	76.2

表 6: The results of Naive Beam Search, ToG methods on CWQ and WebQSP.

B.2 Result Analysis

We conduct a detailed analysis on the answers generated by ToG and ToG-R.

Error Analysis We considered three types of errors: (1) Hallucination error, (2) Refuse error [2], and (3) Format error. The distribution is shown in Figure 6. Our approach has significantly reduced the hallucination and refusal to answer error types in IO and CoT. For GrailQA, ToG even reduces

[2]LLM will refuse to answer due to lack of information.

答案证据 我们对三个数据集中正确回答的样本进行了分析，以调查 LLM 在生成答案时的证据，如图 7 所示。显然，答案中有很大一部分来自 ToG 探索的路径，而大约 20% 的答案完全依赖于嵌入在 LLM 参数中的内在知识来生成。值得注意的是，大约 7% 的正确回答样本需要将探索路径的知识与 LLM 的固有知识结合起来（详见附录表 21）。这一区别使我们的方法不同于传统的基于图的搜索方法，因为它不要求路径完全覆盖包含正确答案的节点。相反，探索路径补充并参考 LLM 的固有知识。ToG-R 的答案类型分布与 ToG 的几乎无法区分，证明了我们方法的鲁棒性。

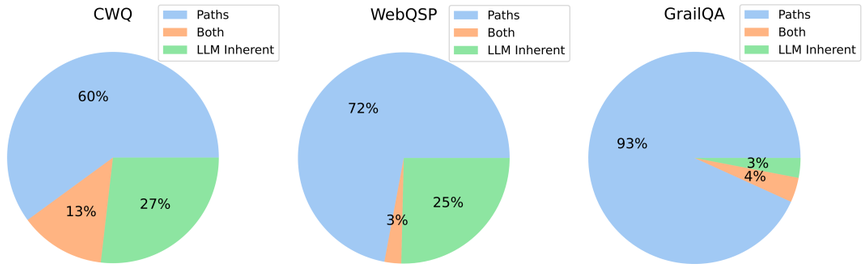

图 7: ToG 在 CWQ、WebQSP 和 GrailQA 数据集上答案证据的比例。

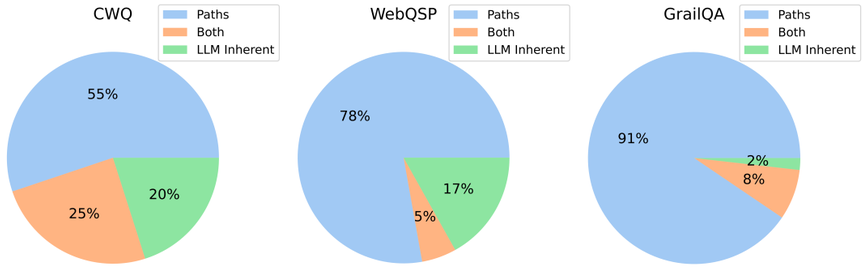

图 8: ToG-R在CWQ、WebQSP和GrailQA数据集上的探索路径重叠率。

探索路径与真实路径之间的重叠比例 我们还对三个数据集中正确回答的样本进行了分析，以研究 ToG 探索路径与 SPARQL 中真实路径之间的重叠比率。重叠比率的定义是重叠路径的数量与真实 SPARQL 中关系总数的比率：

$$\frac{\text{Count}(\text{Rel}(\text{Paths}) \cap \text{Rel}(\text{SPARQL}))}{\text{Count}(\text{Rel}(\text{SPARQL}))}$$

其中 Rel(*) 表示 "*" 中所有不重复的关系，Count(*) 表示 "*" 的数量[3]。图 9 是一条路径示意图，以表 22 中所示案例为例。从图 10 可以观察到，ToG 探索的路径与平均 30% 正确样本的黄金路径完全相同，而平均 21% 正确样本的路径与黄金路径则完全不同。这表明 ToG 成功地探索了知识图谱空间中一条完全且近似全新的路径，以达到最终的答案实体。对于 ToG-R，两者之间的差异主要体现在 CWQ 数据集上，ToG 结果中区间 (25,50] 的百分比相当显著（接近 40%），而 ToG-R 结果则趋于更加均匀分布，如图 11 所示。我们认为这种差异源于对实体的忽视，从而增强了探索关系的多样性。这代表了知识图谱推理在学术研究中的重要应用。

[3]我们通过计算真实 SPARQL 中关系的数量来大致计算路径的长度。

these types of errors by 50% and 60%, respectively. Moreover, in ToG's error samples, there are still many instances of hallucination and refusal to answer errors. This is because the current search depth and width are both set to 3. By increasing the search depth and width, these error instances will further decrease (refer to Section 3.2.3). Furthermore, we currently generalize incorrect answers as hallucinations, but there are various categories within hallucinations, which we won't discuss in this paper. Additionally, after applying ToG, there's a slight increase in samples with format errors. This result shows that the explored paths lead to a noticeable increase in the tokens, sometimes even exceeding the maximum output limit. However, the error rate from this issue is negligible (less than 3%).

Evidence of Answers We conducted an analysis of the correctly answered samples in three datasets to investigate the evidence for LLM in generating answers as shown in Figure 7. Evidently, a significant portion of the answers are derived from the paths explored by ToG, while roughly 20% rely exclusively on the intrinsic knowledge embedded within LLM's parameters for generating responses. It is worth noting that around 7% of the correctly answered samples require a combination of knowledge from both the explored paths and LLM's inherent knowledge (as elaborated in Appendix Table 21). This distinction sets our approach apart from traditional graph-based search methods, as it does not necessitate the path to encompass the node containing the correct answer entirely. Instead, the explored paths supplement and reference LLM's inherent knowledge. The distribution of answer types for ToG-R is almost indistinguishable from that of ToG, proving the robustness of our approach.

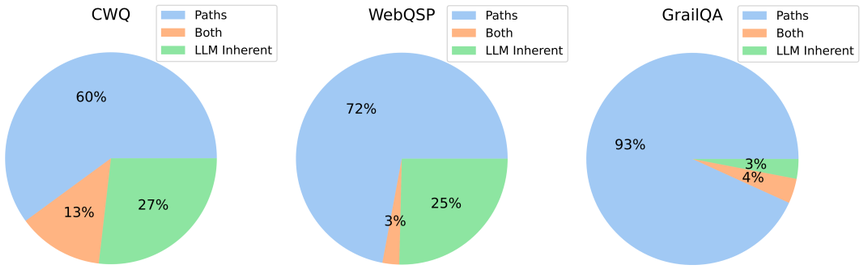

图 7: The proportions of ToG's evidence of answers on CWQ, WebQSP, and GrailQA datasets.

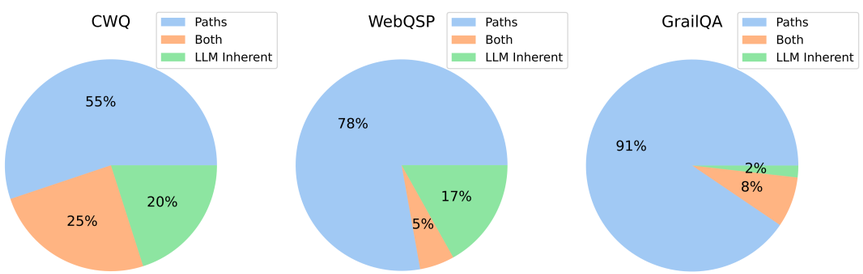

图 8: The explored path overlap ratio of ToG-R on CWQ, WebQSP, and GrailQA datasets.

The Overlap Ratio between the Explored Paths and Ground-truth Paths We also conduct an analysis of the correctly answered samples in three datasets to investigate the ratio of overlap between

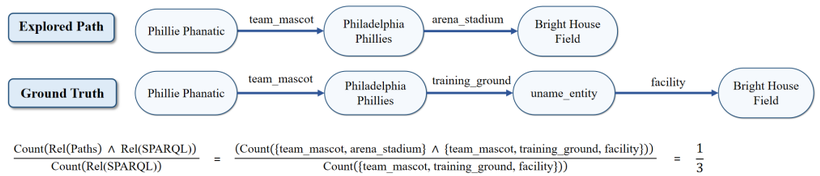

图 9: Path schematic to calculate overlap.

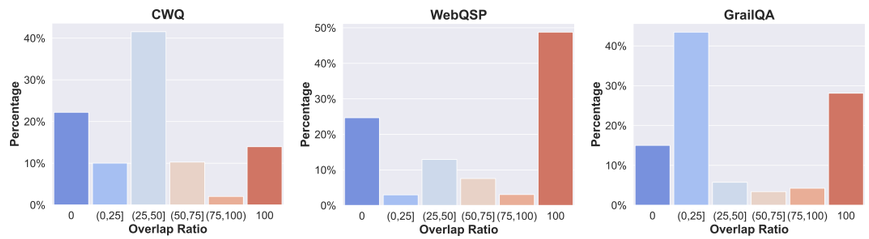

图 10: ToG在CWQ、WebQSP和GrailQA数据集上探索的路径重叠率。

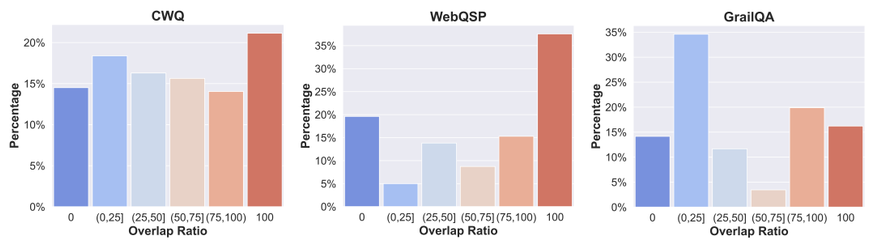

图 11: ToG-R在CWQ、WebQSP和GrailQA数据集上的路径重叠率。

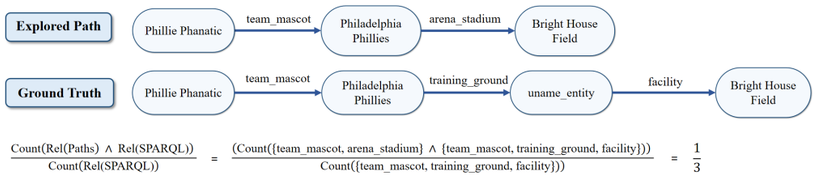

图 9: Path schematic to calculate overlap.

the paths explored by ToG and the ground-truth path in SPARQL. The definition of overlap ratio is the ratio of overlapping paths to the total number of relations in ground-truth SPARQL:

$$\frac{\text{Count}(\text{Rel}(\text{Paths}) \cap \text{Rel}(\text{SPARQL}))}{\text{Count}(\text{Rel}(\text{SPARQL}))}$$

where Rel(*) denotes all the unduplicated relations in the "*" and Count(*) denotes the number of "*"[3]. Figure 9 is a path schematic which takes the case shown in Table 22 for example. It can be observed from Figure 10 that the paths explored by ToG are identical to the golden paths of an average of 30% correct samples, while the paths of an average of 21% correct samples are completely different from the golden path. This indicates that ToG has successfully explored a completely and approximately new path in the knowledge graph space to reach the final answer entity. For ToG-R, the disparity between the two is primarily evident in the CWQ dataset, where the percentage of intervals (25,50] in ToG results is quite significant (nearly 40%), whereas ToG-R results tend to be more evenly distributed as shown in Figure 11. We contend that this discrepancy arises from the disregard of entity, thereby enhancing the diversity of explored relations. This represents a significant application of knowledge graph reasoning in academic research.

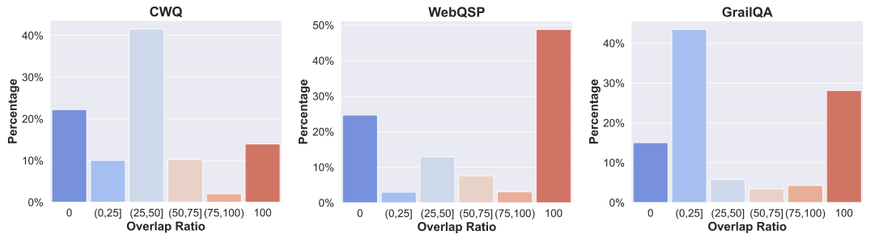

图 10: The explored path overlap ratio of ToG on CWQ, WebQSP, and GrailQA datasets.

The Reasoning Depth of Questions We calculate the reasoning depth of testing questions based on the number of relations within their ground-truth SPARQL queries on CWQ and WebQSP. The counts of questions with different reasoning depths are shown in Figure 12. We analyze the performances of ToG, ToG-R, and CoT on testing questions of both datasets with different reasoning depths. As illustrated in Figure 13, the performances of CoT show roughly decreasing trends on both datasets, with the reasoning depth of testing questions increasing. Conversely, ToG and ToG-R can partially counteract the performance degradation caused by the increment of reasoning depths of questions, especially on CWQ. Generally, the performance difference between ToG and CoT becomes more significant on deeper questions.

[3]We approximately calculate the length of a path by counting the number of relations in the ground-truth SPARQL.

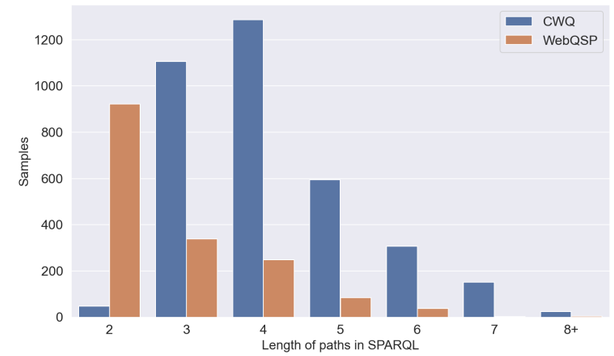

图 12: CWQ 和 WebQSP 数据集中基于关系数量计算的真实 SPARQL 查询的长度。

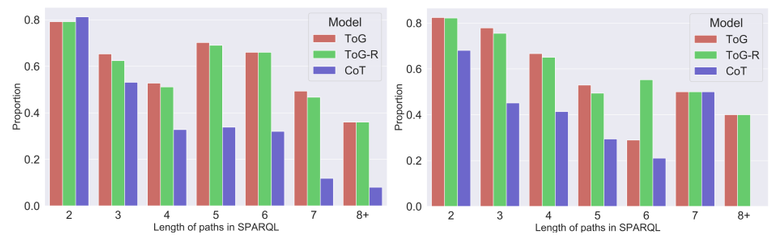

图 13: ToG、ToG-R 和 CoT 在 CWQ 和 WebQSP 数据集上的表现。

问题的推理深度 我们根据CWQ和WebQSP中其真实的SPARQL查询内部关系的数量来计算测试问题的推理深度（reasoning depth）。具有不同推理深度的问题数量如图12所示。我们分析了ToG、ToG-R和CoT在两个数据集上不同推理深度测试问题上的表现。如图 13所示，CoT的表现随着测试问题推理深度的增加而呈现出大致下降的趋势。相反，ToG和ToG-R可以在一定程度上抵消由于问题推理深度增加而导致的性能下降，特别是在CWQ上。总体而言，ToG和CoT之间在深度问题上的性能差异变得更加显著。

B.2.1 EFFICIENCY OF TOG

有许多解决方案可以提高效率，并将ToG的计算复杂度（与调用LLM的数量成比例）从原始的 $O(ND)$ 降低到 $O(D)$，其中 D 是推理路径的深度（或等价于长度），而 N 是束搜索的宽度（每次迭代中池中保留的路径数量）。

解决方案 1 通过在修剪中使用轻量级模型将计算复杂度从 $O(ND)$ 降低到 $O(D)$。计算的瓶颈在于修剪步骤，它导致 $N * D$ 次调用，优化它对于计算效率非常重要。一种技术路线是用小模型（如BM25和Sentence-BERT）替换LLM，因为小模型的调用速度远快于LLM。通过这种方式，我们可以将LLM的调用次数从 $2ND + D + 1$ 减少到 $D + 1$。例如，当 $D = 3$ 时，LLM的调用次数仅为4次。然而，这种优化牺牲了准确性，因为在修剪中使用了较弱的评分模型。例如，如手稿中的表 5 所示，ToG在WebQSP上的表现从76.2%下降到66.3%，这是在将ChatGPT替换为SentenceBERT进行修剪后导致的。为了解决性能下降的问题，我们可以适当地增加搜索宽度来补偿损失，因为增加搜索宽度可以提高在池中选择最佳路径的机会，并且不会影响LLM的调用次数。为了经验验证这一点，我们将搜索宽度从3增加到5，并

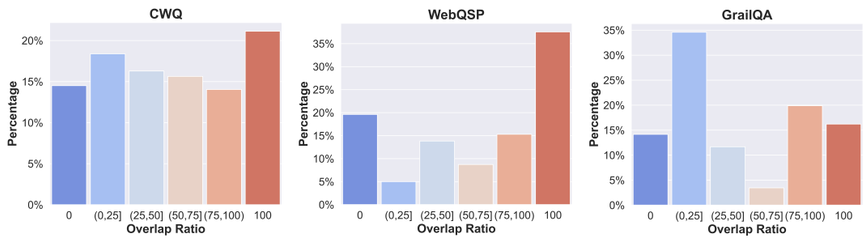

图 11: The path overlap ratio of ToG-R on CWQ, WebQSP, and GrailQA datasets.

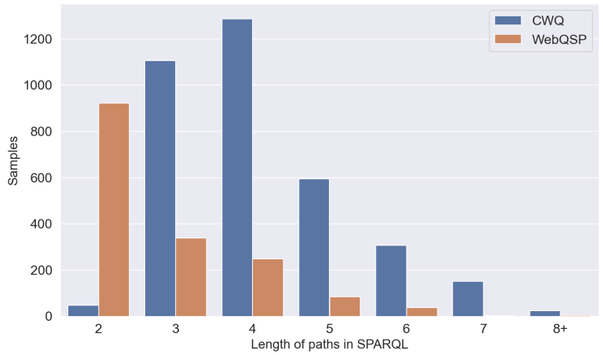

图 12: The lengths of the ground-truth SPARQL queries within the CWQ and WebQSP datasets, computed based on relation numbers.

Dataset	CWQ	WebQSP	GrailQA	QALD10-EN	SimpleQuestion
Average LLM Calls	14.3	11.2	10.4	11.4	8.7

表 7: 每个问题的平均LLM调用次数（第一部分）

在WebQSP上重新评估使用SentenceBERT作为修剪模型的ToG。准确率从66.3%上升到68.5%，并且随着宽度的增加可能会进一步提升，因为更大的宽度不会导致LLM调用次数的增加。

解决方案 2 通过在同一修剪步骤中统一提示，将计算复杂度从 $O(ND)$ 降低到 $O(D)$。加速修剪步骤的另一种解决方案是一次性使用LLM对N个候选集的所有组件进行评分，以获取前N个候选，而不是分别调用LLM N次来单独评分N个候选集。通过这个解决方案，无论是实体修剪步骤还是关系修剪步骤，每次迭代只需要一次LLM调用。因此，ToG和ToG-R每个问题所需的最大LLM调用次数将降至 $2D + D + 1$ 和 $D + D + 1$。

方案 3 优化修剪步骤，使实际调用 LLM 的次数远低于之前估计的 $2ND + D + 1$，并接近一些常见的提示方法，例如 CoT-SC。对于 ToG 和其他基于 LLM 的方法，推理阶段的计算时间（成本或复杂度）主要取决于调用 LLM 的次数。对于每个问题，ToG 最多需要 $2ND + D + 1$ 次。同时，ToG-R 如第 2 节所述，最多需要 $ND + D + 1$ 次。

给定束搜索宽度 N 和最大推理深度 D，ToG 从最与问题中的关键词对齐的实体开始初始化搜索。在推理路径的每个迭代步骤中，ToG 从每个 N 个实体/关系（知识图谱上的节点/边）开始，并搜索其所有邻近的关系/实体。考虑到搜索宽度 N，ToG 始终在池中保留 N 个“最可能”的候选推理路径，因此始终存在 N 个候选实体集合 $E^D_{cand,n}$ 和 N 个候选关系集合 $R^D_{cand,n}$。因此，实体修剪需要 N 次 LLM 调用，关系修剪同样需要 N 次调用，以及一次额外的 LLM 调用用于推理（评估当前候选路径的信息是否足够）。我们必须指出，对于每个 N 个起始实体，其所有邻近实体/关系并不是逐一评分。相反，所有邻近实体/关系被“翻译”成“一”条提示并一并发送到 LLM，LLM 一次性输出前 N 个候选。因此，每个起始实体只需调用一次 LLM 进行修剪，因此 N 个起始实体在一个迭代步骤中总共调用 LLM N 次。因此，推理 D 步后，总共调用次数为 $2ND + D$ 次。最后，还需一次额外调用将最终路径“翻译”成用户可理解的语言并回答用户。因此，ToG 总共需要 $2ND + D + 1$ 次 LLM 调用。由于大多数问题可以在 3 次跳跃内回答（即推理路径深度为 3），而且当搜索宽度 N=3 时，性能通常足够好，如我们在图 3 中测试的那样，总的 LLM 调用次数为 $2 \times 3 \times 3 + 3 + 1 = 22$。因此，计算时间约是仅使用 LLM 时的 21 倍。与 ToG 性能相似，其变种 ToG-R 通过使用随机实体修剪而非基于 LLM 的实体修剪，只需调用 LLM $ND + D + 1$ 次，节省了近一半的计算时间。

$2ND + D + 1$ 是最大计算复杂度。在大多数情况下，ToG 不需要 $2ND + D + 1$ 次 LLM 调用来处理一个问题，因为如果 LLM 确定已检索到足够的信息，整个推理过程可能会在达到最大推理深度 D 之前提前停止。同样，在大多数情况下，ToG-R 确实不需要 $ND + D + 1$ 次 LLM 调用。作为说明，表 7 和表 8 显示了 ToG 在不同数据集上每个问题所需的 LLM 调用的平均次数。可以看出，在四个多跳 KBQA 数据集中，LLM 调用的平均次数（范围从 10 到 15）显著小于 22，这是一种从 $2ND + D + 1$ 计算得出的理论最大 LLM 调用次数，当 N=3 和 D=3 时。我们还可以看到，对于单跳推理数据集，如 SimpleQuestion 和 T-REx，这个平均数甚至更小（< 10）。

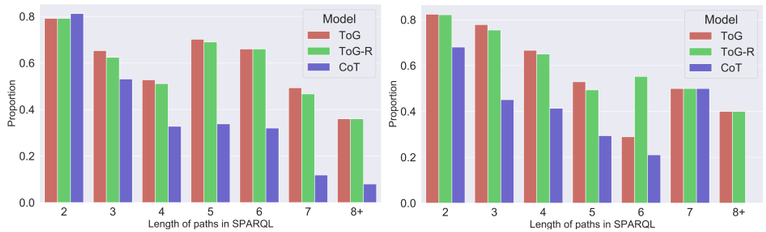

图 13: ToG, ToG-R and CoT's performance among CWQ and WebQSP dataset.

B.2.1 Efficiency of ToG

There are many solutions to improve efficiency and reduce the computational complexity (proportional to the number of calling LLMs) of ToG from the original $O(ND)$ to $O(D)$, where D is the depth (or equivalently length) of the reasoning path, and N is the width of the beam-search (how many paths are remained in the pool in each iteration).

Solution 1 Reducing computational complexity from $O(ND)$ to $O(D)$ by using lightweight model in pruning. The bottleneck of computation is the pruning step, which contributes to $N * D$ times calling, and it is important to optimize it for computational efficiency. A technical route is to replace LLM with small models such as BM25 and Sentence-BERT in the pruning step since the small models are much faster than LLM calling. In this way, we can reduce the number of LLM calling from $2ND + D + 1$ to $D + 1$. When D=3, for example, there are only 4 times LLM calling. However, this optimization sacrifices the accuracy due to the weaker scoring model in pruning. For instance, as shown in Table 5 of the manuscript, the performance of ToG on WebQSP drops from 76.2% to 66.3% after replacing ChatGPT with SentenceBERT for pruning. To alleviate the issue of the performance degradation, we can appropriately increase the search width to compensate the loss because increasing search width can improve the chance of the optimal path to be selected in the pool and it doesn't affect the number of LLM calling. To empirically verify this, we increase the search width from 3 to 5 and reevaluate ToG with SentenceBERT as the pruning model on WebQSP. The accuracy rises to from 66.3% to 68.5% and could be further improved with a greater width since the greater width would not cause an increase in the number of LLM calls.

Solution 2 Reducing computational complexity from $O(ND)$ to $O(D)$ by unifying the prompts in the same pruning step. Another solution on speeding up the pruning step is to employ the LLM at once to score all components of N candidate sets for obtaining top-N candidates, instead of calling the LLM N times to score N candidate sets separately. Through this solution, either entity pruning step or relation pruning step only need 1 LLM call for each iteration. Thus, the maximum number of LLM calls per question needed for ToG and ToG-R would drop to $2D + D + 1$ and $D + D + 1$.

Solution 3 Optimizing pruning step to make the actual calls of LLMs much less than the previously estimated $2ND + D + 1$ and closer to some common prompting methods such as CoT-SC. For ToG and other LLM-based methods, the computational time (cost or complexity) in the inference phase mainly depends on how many times calling LLM. For each question, ToG needs at most $2ND + D + 1$ times. Meanwhile, ToG-R needs at most $ND + D + 1$ times as mentioned in Section 2.

Dataset	WebQuestion	T-REx	Zero-Shot RE	Creak
Average LLM Calls	10.5	7.7	7.6	8.0

表 8: 每个问题的平均大型语言模型（LLM）调用次数（第二部分）

C Dataset

该论文中使用的数据集的统计信息如表 9 所示。我们还提供了每个数据集的详细结果表，见表 10 至表 18，展示了 ToG 相较于之前的基于微调和基于提示的相关工作所带来的增强效果。对于 QALD10-en、WebQuestions、零样本 (Zero-Shot) RE 和 Creak，基于 ChatGPT 的 ToG 达到了新的状态 (state) 的最先进水平。此外，基于 GPT-4 的 ToG 在几乎所有的多跳知识库问答 (Multi-Hop KBQA) 数据集中超越了基于微调的方法，其中在 CWQ 上，ToG 接近最先进水平 (69.5%)。

Dataset	Answer Format	Train	Test	Licence
ComplexWebQuestions	Entity	27,734	3,531	-
WebQSP	Entity/Number	3,098	1,639	CC License
GrailQA*	Entity/Number	44,337	1,000	-
QALD-10	Entity/Number	-	333	MIT License
Simple Quesiton*	Entity/Number	14,894	1,000	CC License
WebQuestions	Entity/Number	3,778	2,032	-
T-REx	Entity	2,284,168	5,000	MIT License
Zero-Shot RE	Entity	147,909	3,724	MIT License
Creak	Bool	10,176	1,371	MIT License

表 9: 本论文中使用的数据集统计。* 表示我们随机选择了 1,000 个样本来自 GrailQA 和 Simple Questions 测试集，以构成测试集，因为测试样本的数量非常丰富。

Dataset	CWQ	WebQSP	GrailQA	QALD10-EN	SimpleQuestion
Average LLM Calls	14.3	11.2	10.4	11.4	8.7

表 7: Average Number of LLM Calls per Question (Part 1)

Dataset	WebQuestion	T-REx	Zero-Shot RE	Creak
Average LLM Calls	10.5	7.7	7.6	8.0

表 8: Average Number of LLM Calls per Question (Part 2)

Given the beam search width N and maximal reasoning depth D, ToG's initialize the search from the entity mostly aligning with the keyword in question. In each iterative step of the reasoning path, ToG starts from each of the N entities/relations (nodes/edges on knowledge graph) and searches all its neighboring relations/entities. Given the search width N, ToG always keep N "most-likely" candidate reasoning paths in the pool, and thus there are always N candidate entity sets $E^D_{cand,n}$ and N candidate relation sets $R^D_{cand,n}$. Consequently, it needs N LLM calls for entity pruning and N calls for relation pruning, respectively, as well as one additional LLM call for reasoning (evaluating if the information from the current candidate paths are enough or not). We have to point it out that, for each of the N starting entities, all its neighbor entities/relations are NOT scored one by one. On the contrary, all its neighbor entities/relations are "translated" into "one" prompt altogether and are sent to LLM, which output the top-N candidates at one-time. Therefore, each starting entity only calls LLM once for pruning and so N starting entities calls LLM N times in one iterative step. Consequently, there are totally $2ND + D$ times calling after reasoning D steps. In the end, there is an additional calling that "translate" the final path to user-understandable language and answer the user. Therefore, ToG requires $2ND + D + 1$ LLM calls in total. Since most questions can be answered within 3 hops (means depth of reasoning path is 3), and the performance is usually good enough when the search width N=3 as we tested in Figure 3, the total number of LLM calling is $2 \times 3 \times 3 + 3 + 1 = 22$. So the computational time is about 21 times longer than that of LLM-only. With a similar performance to ToG, its variant ToG-R only calls LLM for $ND + D + 1$ times by using random entity pruning instead of LLM-based entity pruning, saving nearly half of computational time.

$2ND + D + 1$ is the maximal computational complexity. In most cases, ToG does not need $2ND + D + 1$ LLM calls for a question because the whole reasoning process might be early stopped before the maximum reasoning depth D is reached if LLM determines enough information has been retrieved. Likewise, ToG-R does not really need $ND + D + 1$ LLM calls in most cases. As an illustration, Table 7 and Table 8 show the average numbers of LLM calls per question needed by ToG on different datasets. It can be seen that in the four multi-hop KBQA datasets, the average numbers of LLM calls (ranging from 10 to 15) are significantly smaller than 22, which is the theoretical maximum number of LLM calls calculated from $2ND + D + 1$ when N=3 and D=3. We can also see that this AVERAGE number gets even smaller (< 10) for single-hop reasoning datasets, such as SimpleQuestion and T-REx.

C DATASET

The statistics of the datasets used in this paper are shown in Table 9. We also provide a detailed result table for each dataset, shown in Table 10 to Table 18, illustrating the enhancements of ToG

Model	**Method**	**EM**
Fine-Tuning	QGG (Query Graph Generator) (Lan & Jiang, 2020)	44.1
	PullNet (Sun et al., 2019)	45.9
	NSM+h (He et al., 2021)	53.9
	CBR-KBQA (Das et al., 2021)	67.1
	DecAF (Yu et al., 2023)	70.4
ChatGPT	KD-CoT (Wang et al., 2023b)	49.2
	ToG	57.1
	ToG-R	**58.9**
Llama2-70B-Chat	ToG	53.6
	ToG-R	**57.6**
GPT-4	ToG	67.6
	ToG-R	**69.5**

表 10: 复杂网络问题数据集的微调（Fine-Tuning）和基于提示（prompting）方法的静态分析。

Model	**Method**	**EM**
Fine-Tuning	KD-CoT (Wang et al., 2023b)	73.7
	NSM (He et al., 2021)	74.3
	Program Transfer (Cao et al., 2022)	74.6
	TIARA (Shu et al., 2022)	75.2
	DecAF (Yu et al., 2023)	82.1
Code-davinci-002	KB-BINDER (Li et al., 2023a)	74.4
ChatGPT	StructGPT (Jiang et al., 2023)	72.6
	ToG-R	75.8
	ToG	**76.2**
Llama2-70B-Chat	ToG-R	**69.4**
	ToG	64.1
GPT-4	ToG-R	81.9
	ToG	**82.6**

表 11: WebQSP 数据集的微调静态分析（Fine-Tuning）及基于提示（prompting）的方法。

Model	**Method**	**EM**
Fine-Tuning	DecAF (Yu et al., 2023)	68.4
	UniParser (Liu et al., 2022)	69.5
	TIARA (Shu et al., 2022)	73.0
	Pangu (Gu et al., 2023)	75.4
Code-davinci-002	KB-BINDER (Li et al., 2023a)	53.2
ChatGPT	ToG-R	66.4
	ToG	**68.7**
GPT-4	ToG-R	80.3
	ToG	**81.4**

表 12: GrailQA 数据集的微调静态（Fine-Tuning）和基于提示（prompting-based）的方法

compared to the previous fine-tuning-based and prompting-based relevant works. For QALD10-en, WebQuestions, Zero-Shot RE, and Creak, ChatGPT-based ToG reached a new state-of-the-art. Furthermore, GPT-4-based ToG exceeded the fine-tuning-based approaches on almost all Multi-Hop KBQA datasets, where on CWQ, ToG is close to the state-of-the-art (69.5%).

Dataset	Answer Format	Train	Test	Licence
ComplexWebQuestions	Entity	27,734	3,531	-
WebQSP	Entity/Number	3,098	1,639	CC License
GrailQA*	Entity/Number	44,337	1,000	-
QALD-10	Entity/Number	-	333	MIT License
Simple Quesiton*	Entity/Number	14,894	1,000	CC License
WebQuestions	Entity/Number	3,778	2,032	-
T-REx	Entity	2,284,168	5,000	MIT License
Zero-Shot RE	Entity	147,909	3,724	MIT License
Creak	Bool	10,176	1,371	MIT License

表 9: The statistics of the datasets used in this paper. * denotes we randomly selected 1,000 samples from the GrailQA and Simple Questions test set to constitute the testing set owing to the abundance of test samples.

Model	Method	Acc
Fine-Tuning	SPARQL-QA(Santana et al., 2022)	45.4
ChatGPT	ToG-R	48.6
	ToG	**50.2**
GPT-4	ToG	53.8
	ToG-R	**54.7**

表 13: QALD10-en 数据集的微调（Fine-Tuning）静态（statics）和基于提示（prompting-based）的方法。

Model	Method	EM
Fine-Tuning	T5-LARGE+KPs (dos Santos et al., 2022)	58.3
	Memory Networks (Bordes et al., 2015)	63.9
	GETT-QA (Banerjee et al., 2023)	76.1
	DiFaR(Baek et al., 2023a)	**85.8**
ChatGPT	ToG-R	45.4
	ToG	**53.6**
GPT-4	ToG-R	58.6
	ToG	**66.7**

表 14: 简单问题数据集的微调（Fine-Tuning）静态分析，基于提示（prompting）的方法。

Model	Method	EM
Fine-Tuning	T5.1.1-XXL+SSM (Raffel et al., 2020)	43.5
	PaLM (Chowdhery et al., 2022)	43.5
	RAG (Lewis et al., 2021)	45.2
	FiDO (de Jong et al., 2022)	51.1
	FiE+PAQ (Kedia et al., 2022)	56.3
PALM2	Few-shot (Li et al., 2023a)	28.2
ChatGPT	$\text{BeamSearchQA}_{\text{Fine-tuned Retriever}}$ (Sun et al., 2023a)	27.3
	ToG-R	53.2
	ToG	**54.5**
GPT-4	ToG-R	57.1
	ToG	**57.9**

表 15: WebQuestions 数据集的微调（Fine-Tuning）静态（statics）和基于提示（prompting-based）的方法。

Model	Method	EM
Fine-Tuning	QGG (Query Graph Generator) (Lan & Jiang, 2020)	44.1
	PullNet (Sun et al., 2019)	45.9
	NSM+h (He et al., 2021)	53.9
	CBR-KBQA (Das et al., 2021)	67.1
	DecAF (Yu et al., 2023)	70.4
ChatGPT	KD-CoT (Wang et al., 2023b)	49.2
	ToG	57.1
	ToG-R	**58.9**
Llama2-70B-Chat	ToG	53.6
	ToG-R	**57.6**
GPT-4	ToG	67.6
	ToG-R	**69.5**

表 10: The statics of Fine-Tuning, prompting-based methods of ComplexWebQuestions dataset.

Model	Method	EM
Fine-Tuning	KD-CoT (Wang et al., 2023b)	73.7
	NSM (He et al., 2021)	74.3
	Program Transfer (Cao et al., 2022)	74.6
	TIARA (Shu et al., 2022)	75.2
	DecAF (Yu et al., 2023)	82.1
Code-davinci-002	KB-BINDER (Li et al., 2023a)	74.4
ChatGPT	StructGPT (Jiang et al., 2023)	72.6
	ToG-R	75.8
	ToG	**76.2**
Llama2-70B-Chat	ToG-R	**69.4**
	ToG	64.1
GPT-4	ToG-R	81.9
	ToG	**82.6**

表 11: The statics of Fine-Tuning, prompting-based methods of WebQSP dataset.

Model	Method	EM
Fine-Tuning	DecAF (Yu et al., 2023)	68.4
	UniParser (Liu et al., 2022)	69.5
	TIARA (Shu et al., 2022)	73.0
	Pangu (Gu et al., 2023)	75.4
Code-davinci-002	KB-BINDER (Li et al., 2023a)	53.2
ChatGPT	ToG-R	66.4
	ToG	**68.7**
GPT-4	ToG-R	80.3
	ToG	**81.4**

表 12: The statics of Fine-Tuning, prompting-based methods of GrailQA dataset

Model	Method	EM
Fine-Tuning	MetaRAG	78.7
	Wikipedia	81.3
	single ngram	83.7
	KGI_1	84.4
	Re2G (Glass et al., 2022)	87.7
ChatGPT	ToG-R	75.3
	ToG	**76.8**
GPT-4	ToG-R	75.5
	ToG	**77.1**

表 16: T-REx 数据集的微调统计（Fine-Tuning），基于提示的方法（prompting-based methods），数据来自排行榜（leaderboard）。

Model	Method	EM
Fine-Tuning	Multitask DPR + BART	58.0
	MetaRAG	71.6
	KGI_1	72.6
	Wikipedia	74.0
	single ngram	74.6
ChatGPT	ToG-R	86.5
	ToG	**88.0**
GPT-4	ToG-R	86.9
	ToG	**88.3**

表 17: 零样本 (Zero-Shot) 关系抽取 (RE) 的精调 (Fine-Tuning) 和基于提示 (prompting-based) 方法的统计数据，其中数据来自排行榜 (leaderboard)。

Model	Method	EM
Fine-Tuning	RoBERTa-Large (Liu et al., 2019)	80.6
	T5-3B (Raffel et al., 2020)	85.6
	RACo-Large (Yu et al., 2022)	88.2
ChatGPT	ToG-R	**93.8**
	ToG	91.2
GPT-4	ToG-R	95.4
	ToG	**95.6**

表 18: Creak 数据集的微调 (Fine-Tuning) 静态分析。

Model	Method	Acc
Fine-Tuning	SPARQL-QA(Santana et al., 2022)	45.4
ChatGPT	ToG-R	48.6
	ToG	**50.2**
GPT-4	ToG	53.8
	ToG-R	**54.7**

表 13: The statics of Fine-Tuning, prompting-based methods of QALD10-en dataset.

Model	Method	EM
Fine-Tuning	T5-LARGE+KPs (dos Santos et al., 2022)	58.3
	Memory Networks (Bordes et al., 2015)	63.9
	GETT-QA (Banerjee et al., 2023)	76.1
	DiFaR(Baek et al., 2023a)	**85.8**
ChatGPT	ToG-R	45.4
	ToG	**53.6**
GPT-4	ToG-R	58.6
	ToG	**66.7**

表 14: The statics of Fine-Tuning, prompting-based methods of SimpleQuetsions dataset.

Model	Method	EM
Fine-Tuning	T5.1.1-XXL+SSM (Raffel et al., 2020)	43.5
	PaLM (Chowdhery et al., 2022)	43.5
	RAG (Lewis et al., 2021)	45.2
	FiDO (de Jong et al., 2022)	51.1
	FiE+PAQ (Kedia et al., 2022)	56.3
PALM2	Few-shot (Li et al., 2023a)	28.2
ChatGPT	$\text{BeamSearchQA}_{\text{Fine-tuned Retriever}}$ (Sun et al., 2023a)	27.3
	ToG-R	53.2
	ToG	**54.5**
GPT-4	ToG-R	57.1
	ToG	**57.9**

表 15: The statics of Fine-Tuning, prompting-based methods of WebQuestions dataset.

D CASE STUDY

在本节中，我们展示CWQ数据集的案例分析，以评估ToG的实用性和局限性。我们将ToG与IO、CoT和新必应搜索引擎进行了比较[4]。我们选择了四个示例进行分析，每个示例都包含前3个推理路径和标准化分数。

在表19中的第一个示例中，ToG最初识别了问题中的“Arthur Miller”和“Lucian”，随后通过探索（exploration）和推理（reasoning）过程扩展其推理路径。经过两次搜索迭代，ToG成功找到了正确答案，因为它通过推理路径将两个实体连接起来，这代表了寻找解决方案的完美路径。此外，推理路径中间步骤的*UnName_Entity*的存在反映出知识图谱的不完整性（即，某些实体缺少“名称”关系）。然而，ToG仍然能够进行下一步推理，因为所有可用关系都包含相关信息。我们观察到，IO和CoT未能正确回答查询，因为它们缺乏适当的知识，而新必应在检索过程中未能检索到适当的信息。

在表20中显示的第二个示例中，IO提示、CoT甚至新必应都出现了幻觉问题，提供了错误答案“Florida”，因为“Renegade”是“Florida State Seminoles”的吉祥物，而不是“战歌”。ToG获得的推理路径是“Renegade” → “sports.fight_song.sports_team” → “Pittsburgh Steeler”。然而，这条推理路径并没有导致最终答案，但结合LLMs，ToG可以回答正确答案“Pennsylvania”。

表21中的第三个示例展示了ToG-R的一个例子，在这个例子中，ToG忽略了中间实体，而是专注于关系中的信息。在推理经过两跳到达“Harvard College”后，结合LLMs，ToG给出了最终结果：“Massachusetts”。可以观察到，IO和CoT没有背景知识，而新必应能够正确回答问题，因为它检索到了正确的信息。

最后一个示例显示在表22中。在这里，ToG生成了到最终问题的推理路径（路径1）。值得注意的是，答案的真实推理路径是*sports.sports_team.team_mascot* → *base.schemastaging.team_training_ground_relationship.facility* → *base.schemastaging.sports_team_extra.training_ground*（可以从SPARQL中检索），其跳数比ToG更多。ToG使探索新的推理路径以达到正确答案成为可能，这代表了知识图谱推理的重要应用。然而，当前知识库（KB）中问题的答案是“Bright House Field”，这是错误的，因为“Philadelphia Phillies”的训练场现在是“Spectrum Field”。这个例子体现了ToG的一个限制，特别是其对知识库准确性的依赖，错误的知识库对ToG的推理准确性产生负面影响。然而，如图4所示，ToG提供了一种新颖的框架，以构建知识图谱的自动化知识注入。

[4]访问版本为2023年7月。

Model	Method	EM
Fine-Tuning	MetaRAG	78.7
	Wikipedia	81.3
	single ngram	83.7
	KGI_1	84.4
	Re2G (Glass et al., 2022)	87.7
ChatGPT	ToG-R	75.3
	ToG	**76.8**
GPT-4	ToG-R	75.5
	ToG	**77.1**

表 16: The statics of Fine-Tuning, prompting-based methods of T-REx dataset, where data are from the leaderboard.

Model	Method	EM
Fine-Tuning	Multitask DPR + BART	58.0
	MetaRAG	71.6
	KGI_1	72.6
	Wikipedia	74.0
	single ngram	74.6
ChatGPT	ToG-R	86.5
	ToG	**88.0**
GPT-4	ToG-R	86.9
	ToG	**88.3**

表 17: The statics of Fine-Tuning, prompting-based methods of Zero-Shot RE, where data are from the leaderboard.

Model	Method	EM
Fine-Tuning	RoBERTa-Large (Liu et al., 2019)	80.6
	T5-3B (Raffel et al., 2020)	85.6
	RACo-Large (Yu et al., 2022)	88.2
ChatGPT	ToG-R	**93.8**
	ToG	91.2
GPT-4	ToG-R	95.4
	ToG	**95.6**

表 18: The statics of Fine-Tuning, prompting-based methods of Creak dataset.

Question	Who influenced Arthur Miller that was influenced by Lucian?
Reasoning Paths	Arthur Miller → *influence.influence_node.influenced_by* → William Shakespeare → *influence.influence_node.influenced_by* → Lucian. (**Path 1**, Score: 0.75) Lucian → *influence.influence_node.influenced_by* → Socrates → *influence.influence_node.influenced_by* → Parmenides. (**Path 2**, Score: 0.2) Arthur Miller → *people.person.education* → **UnName_Entity** → *education.education.student* → Arthur Miller. (**Path 3**, Score: 0.05)
Outputs	**IO**: Edmund Wilson. **CoT**: Arthur Miller was influenced by the playwright and director Thornton Wilder, who in turn was influenced by Lucian of Samosata. **New Bing**: I'm not sure about Lucian's influence on Arthur Miller. However, Arthur Miller was influenced by Henrik Ibsen and Eugene O'Neill. He was also influenced by the events of the McCarthy era and the Salem witch trials. **ToG**: Based on the given knowledge triples, we can infer that Arthur Miller was influenced by William Shakespeare, who was directly influenced by Lucian.
Ground Truth	William Shakespeare.

表 19: 案例用于 ToG。 错误答案 用红色突出显示，而 正确答案 用蓝色突出显示。

Question	What is the state where the team whose fight song is "Renegade" is from?
Reasoning Paths	Renegade → *sports.fight_song.sports_team* → Pittsburgh Steelers. (**Path 1**, Score: 0.67) Renegade → *sports.sports_team.fight_song* → **UnName_Entity**. (**Path 2**, Score: 0.33)
Outputs	**IO**: Florida. **CoT**: The team whose fight song is "Renegade" is the Florida State Seminoles, and they are from the state of Florida in the United States. **New Bing**: The team whose fight song is "Renegade" is the Florida State Seminoles. They are from the state of Florida in the United States. **ToG**: Based on the given knowledge triples, the team with the fight song "Renegade" is the Pittsburgh Steelers. The Pittsburgh Steelers are a professional American football team, and they are from the state of Pennsylvania.
Ground Truth	Pittsburgh Pennsylvania.

表 20: 案例展示 错误答案 用红色高亮，正确答案 用蓝色高亮。

D CASE STUDY

In this section, we present a case analysis of the CWQ dataset to evaluate the utility and limitations of the ToG. We compared ToG with IO, CoT and the New Bing search engine[4]. We have selected four examples for analysis, each with top-3 reasoning paths and normalized scores.

In the first example in Table 19, ToG initially identifies "Arthur Miller" and "Lucian", in the question and subsequently expands its reasoning path through the `Exploration` and `Reasoning` processes. After conducting two iterations of the search, ToG successfully arrived at the correct answer, as it links the two entities with the reasoning path, which represents the perfect route for locating solutions. Additionally, the presence of *UnName_Entity* in the intermediate steps of reasoning paths, reflects the incompleteness of the knowledge graph (i.e., some entities lack the "name" relation). However, ToG is still capable of performing the next reasoning step, as all available relations contain relevant information. We observe that IO and CoT do not answer the query correctly since they lack the appropriate knowledge, and New Bing do not retrieve the appropriate information during the retrieval process.

In the second example shown in Table 20, IO prompt and CoT even New Bing suffer from a hallucination issue and provide an erroneous answer, "Florida", since the "Renegade" is the mascot of "Florida State Seminoles" instead of "fight song". ToG obtain the reasoning path "Renegade" $\rightarrow$ "sports.fight_song.sports_team" $\rightarrow$ "Pittsburgh Steeler". However, this reasoning path does not lead to a final answer, but combined with LLMs', ToG can answer the correct answer "Pennsylvania".

The third example in Table 21 demonstrates an example of the ToG-R, where ToG ignores the intermediate entities and focuses on the information in the relations instead. After two-hop of reasoning to "Harvard College", combined with LLMs', ToG gives the final result: "Massachusetts". It can be observed that IO and CoT do not have background knowledge, and New Bing answers the question correctly since it retrieves the correct information.

The final example is shown in Table 22. Where ToG generates a reasoning path to the final question (Path 1). Notably, the Ground-Truth reasoning path for the answer is *sports.sports_team.team_mascot* $\rightarrow$ *base.schemastaging.team_training_ground_relationship.facility* $\rightarrow$ *base.schemastaging.sports_team_extra.training_ground* (retrievable from the SPARQL), which is more hop than ToG. The ToG enables the exploration of new reasoning paths to reach the correct answer, which represents a significant application of knowledge graph reasoning. However, the answer to the current question in the KB, is "Bright House Field", which is incorrect since "Philadelphia Phillies" training stadium is "Spectrum Field" now. This example exemplifies a constraint of ToG, specifically its dependence on the correctness of the KB, where the incorrect KB has negative impact on ToG's reasoning accuracy. However, as depicted in Figure 4, ToG presents a novel framework to construct automated knowledge infusion to the KG.

[4]Accessed version July 2023.

Question	What state is the college that Sampson Salter Blowers is a grad student of located?
Reasoning Paths	Sampson Salter Blowers $\rightarrow$ *education.education.student* $\rightarrow$ *education.education.institution* $\rightarrow$ {Harvard College}. (**Path 1**, Score: 0.75) Sampson Salter Blowers $\rightarrow$ *education.education.student* $\rightarrow$ *education.educational_institution.students_graduates* $\rightarrow$ {}. (**Path 2**, Score: 0.2) Sampson Salter Blowers $\rightarrow$ *education.education.student* $\rightarrow$ *people.person.education* $\rightarrow$ {}. (**Path 3**, Score: 0.05)
Outputs	**IO**: Unknown. **CoT**: I'm sorry, but I do not have enough information to answer your question. Can you please provide more context or details about Sampson Salter Blowers and the college he attended? **New Bing**: I'm sorry but I couldn't find any information about Sampson Salter Blowers being a grad student. However, he was a noted North American lawyer, Lovalist and jurist from Nova Scotia who graduated from Harvard College with a ba in 1763 and an ma in 1765. He was admitted as an attorney in the Suffolk lnferior Court in 1766. **ToG**: Based on the given relations chains, Sampson Salter Blowers is a grad student of Harvard College. Therefore, the answer to the question is Harvard College, which is located in Massachusetts.
Ground Truth	Massachusetts.

表 21: ToG-R 的案例。"未知"表示 LLM (大语言模型) 无法生成有效的响应。错误答案用红色高亮显示，正确答案用蓝色高亮显示。

Question	Who influenced Arthur Miller that was influenced by Lucian?
Reasoning Paths	Arthur Miller → *influence.influence_node.influenced_by* → William Shakespeare → *influence.influence_node.influenced_by* → Lucian. (**Path 1**, Score: 0.75) Lucian → *influence.influence_node.influenced_by* → Socrates → *influence.influence_node.influenced_by* → Parmenides. (**Path 2**, Score: 0.2) Arthur Miller → *people.person.education* → **UnName_Entity** → *education.education.student* → Arthur Miller. (**Path 3**, Score: 0.05)
Outputs	**IO**: Edmund Wilson. **CoT**: Arthur Miller was influenced by the playwright and director Thornton Wilder, who in turn was influenced by Lucian of Samosata. **New Bing**: I'm not sure about Lucian's influence on Arthur Miller. However, Arthur Miller was influenced by Henrik lbsen and Eugene O'Neill. He was also influenced by the events of the McCarthy era and the Salem witch trials. **ToG**: Based on the given knowledge triples, we can infer that Arthur Miller was influenced by William Shakespeare, who was directly influenced by Lucian.
Ground Truth	William Shakespeare.

表 19: Case for ToG. The wrong answers are highlighted with red color, and the correct answers are highlighted with blue color.

Question	What is mascot Phillie Phanatic's team's spring training stadium?
Reasoning Paths	Phillie Phanatic → *sports.sports_team.team_mascot* → Philadelphia Phillies → *sports.sports_team.arena_stadium* → Bright House Field. (**Path 1**, Score: 0.8) Phillie Phanatic → *sports.sports_team.team_mascot* → Philadelphia Phillies → *ports.sports_team_location.teams* → Philadelphia. (**Path 2**, Score: 0.15) Phillie Phanatic → *sports.sports_team.team_mascot* → Philadelphia Phillies → *sports.sports_team.team* → Phillie Phanatic. (**Path 3**, Score: 0.05)
Outputs	**IO**: Spectrum Field. **CoT**: First, the mascot Phillie Phanatic represents the Philadelphia Phillies baseball team. Second, the Philadelphia Phillies' spring training stadium is located in Clearwater, Florida and is known as Spectrum Field. The answer is Spectrum Field. **New Bing**: The Philadelphia Phillies' spring training stadium is called BayCare Ballpark. **ToG**: The spring training stadium for Phillie Phanatic's team, the Philadelphia Phillies, is Bright House Field.
Ground Truth	Bright House Field (KB), Spectrum Field (Actual).

表 22: 案例对于 ToG。 错误答案 以红色突出显示， 正确答案 以蓝色突出显示。

Question	What is the state where the team whose fight song is "Renegade" is from?
Reasoning Paths	Renegade → *sports.fight_song.sports_team* → Pittsburgh Steelers. (**Path 1**, Score: 0.67) Renegade → *sports.sports_team.fight_song* → **UnName_Entity**. (**Path 2**, Score: 0.33)
Outputs	**IO**: Florida. **CoT**: The team whose fight song is "Renegade" is the Florida State Seminoles, and they are from the state of Florida in the United States. **New Bing**: The team whose fight song is "Renegade" is the Florida State Seminoles. They are from the state of Florida in the United States. **ToG**: Based on the given knowledge triples, the team with the fight song "Renegade" is the Pittsburgh Steelers. The Pittsburgh Steelers are a professional American football team, and they are from the state of Pennsylvania.
Ground Truth	Pittsburgh Pennsylvania.

表 20: Case for ToG. The wrong answers are highlighted with red color, and the correct answers are highlighted with blue color.

E SPARQL AND PROMPTS

In this section, we show all the prompts that need to be used in the main experiments. First, we pre-define SPARQL for Freebase queries, which can be executed by simply filling in the appropriate mid and relation. For Wikidata, we abstain from employing executable SPARQL, rather we directly engage in querying through nine pre-defined service APIs.

E.1 PRE-DEFINED SPARQL

E.1.1 RELATION SEARCH

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?relation
WHERE {
    ns:mid ?relation ?x .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?relation
WHERE {
    ?x ?relation ns:mid .
}
```


E SPARQL AND PROMPTS

在这一部分中，我们展示了在主要实验中需要使用的所有提示。首先，我们预定义了用于 Freebase 查询的 SPARQL（SPARQL），可以通过简单地填写适当的 mid 和关系来执行。对于 Wikidata，我们不使用可执行的 SPARQL，而是直接通过九个预定义的服务 API 进行查询。

E.1 PRE-DEFINED SPARQL

E.1.1 RELATION SEARCH

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?relation
WHERE {
    ns:mid ?relation ?x .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?relation
WHERE {
    ?x ?relation ns:mid .
}
```

E.1.2 ENTITY SEARCH

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
    ns:mid ns:relation ?tailEntity .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
    ?tailEntity ns:mid ns:relation  .
}
```

E.1.3 CONVERT MID TO LABEL

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT DISTINCT ?tailEntity
WHERE {
{
    ?entity ns:type.object.name ?tailEntity .
    FILTER(?entity = ns:mid)
}
UNION
{
    ?entity <http://www.w3.org/2002/07/owlsameAs> ?tailEntity .
```

Question	What state is the college that Sampson Salter Blowers is a grad student of located?
Reasoning Paths	Sampson Salter Blowers → *education.education.student* → *education.education.institution* → {Harvard College}. (**Path 1**, Score: 0.75) Sampson Salter Blowers → *education.education.student* → *education.educational_institution.students_graduates* → {}. (**Path 2**, Score: 0.2) Sampson Salter Blowers → *education.education.student* → *people.person.education* → {}. (**Path 3**, Score: 0.05)
Outputs	**IO**: Unknown. **CoT**: I'm sorry, but I do not have enough information to answer your question. Can you please provide more context or details about Sampson Salter Blowers and the college he attended? **New Bing**: I'm sorry but I couldn't find any information about Sampson Salter Blowers being a grad student. However, he was a noted North American lawyer, Lovalist and jurist from Nova Scotia who graduated from Harvard College with a ba in 1763 and an ma in 1765. He was admitted as an attorney in the Suffolk lnferior Court in 1766. **ToG**: Based on the given relations chains, Sampson Salter Blowers is a grad student of Harvard College. Therefore, the answer to the question is Harvard College, which is located in Massachusetts.
Ground Truth	Massachusetts.

表 21: Case for ToG-R. "Unknown" denotes LLM is unable to generate a valid response. The wrong answers are highlighted with red color, and the correct answers are highlighted with blue color.


```
    FILTER(?entity = ns:mid)
}
}
```

E.2 Pre-defined APIs

```
def label2qid(self, label: str) -> str:

def label2pid(self, label: str) -> str:

def pid2label(self, pid: str) -> str:

def qid2label(self, qid: str) -> str:

def get_all_relations_of_an_entity(self, entity_qid: str)
    -> tp.Dict[str, tp.List]:

def get_tail_entities_given_head_and_relation(self, head_qid: str,
    relation_pid: str)
    -> tp.Dict[str, tp.List]:

def get_tail_values_given_head_and_relation(self, head_qid: str,
    relation_pid: str) -> tp.List[str]:

def get_external_id_given_head_and_relation(self, head_qid: str,
    relation_pid: str) -> tp.List[str]:

def mid2qid(self, mid: str) -> str:
```

E.3 ToG

E.3.1 Relation Prune

请检索 k 个对问题有贡献的关系（用分号分隔），并按0到1的比例对它们的贡献进行评分（k 个关系的分数之和为1）。

上下文少样本 (In-Context Few-shot)

问：{查询}

主题实体：{主题实体}

关系：{关系列表}

答：

E.3.2 Entity Prune

Question	What is mascot Phillie Phanatic's team's spring training stadium?
Reasoning Paths	Phillie Phanatic → *sports.sports_team.team_mascot* → Philadelphia Phillies → *sports.sports_team.arena_stadium* → Bright House Field. (**Path 1**, Score: 0.8) Phillie Phanatic → *sports.sports_team.team_mascot* → Philadelphia Phillies → *ports.sports_team_location.teams* → Philadelphia. (**Path 2**, Score: 0.15) Phillie Phanatic → *sports.sports_team.team_mascot* → Philadelphia Phillies → *sports.sports_team.team* → Phillie Phanatic. (**Path 3**, Score: 0.05)
Outputs	**IO**: Spectrum Field. **CoT**: First, the mascot Phillie Phanatic represents the Philadelphia Phillies baseball team. Second, the Philadelphia Phillies' spring training stadium is located in Clearwater, Florida and is known as Spectrum Field. The answer is Spectrum Field. **New Bing**: The Philadelphia Phillies' spring training stadium is called BayCare Ballpark. **ToG**: The spring training stadium for Phillie Phanatic's team, the Philadelphia Phillies, is Bright House Field.
Ground Truth	Bright House Field (KB), Spectrum Field (Actual).

表 22: Case for ToG. The wrong answers are highlighted with red color, and the correct answers are highlighted with blue color.

请在0到1的范围内为实体对问题的贡献评分（所有实体的评分总和为1）。

上下文少样本（few-shot）

问：{查询}

关系：{当前关系}

实体：{实体列表}

评分：

E.3.3 REASONING

给定一个问题和相关的检索知识图谱三元组（实体，关系，实体），你需要回答这些三元组和你的知识是否足以回答这个问题（是或否）。

上下文少样本

Q: {查询}

知识三元组: {探索路径}

A:

E.3.4 GENERATE

给定一个问题和相关的检索知识图谱三元组（实体，关系，实体），你需要用这些三元组和你自己的知识来回答这个问题。

上下文少样本

Q: {查询}

知识三元组: {探索路径}

A:

E.4 TOG-R

E.4.1 REASONING

请使用主题实体（Topic Entity）、关系链（Relations Chains）及其候选实体（Candidate Entities）回答问题，您需要回答使用这些三元组和您的知识是否足够回答该问题（是或否）。

上下文少样本

Q: {查询}

主题实体（Topic Entity），带有关系链（relations chains）及其候选实体（candidate entities）：{探索的关系链（Explored Relation Chains）}

A:

E.1.2 ENTITY SEARCH

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
    ns:mid ns:relation ?tailEntity .
}

PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
    ?tailEntity ns:mid ns:relation  .
}
```

E.1.3 CONVERT MID TO LABEL

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT DISTINCT ?tailEntity
WHERE {
{
    ?entity ns:type.object.name ?tailEntity .
    FILTER(?entity = ns:mid)
}
UNION
{
    ?entity <http://www.w3.org/2002/07/owlsameAs> ?tailEntity .
    FILTER(?entity = ns:mid)
}
}
```

E.2 PRE-DEFINED APIS

```
def label2qid(self, label: str) -> str:

def label2pid(self, label: str) -> str:

def pid2label(self, pid: str) -> str:

def qid2label(self, qid: str) -> str:

def get_all_relations_of_an_entity(self, entity_qid: str)
    -> tp.Dict[str, tp.List]:

def get_tail_entities_given_head_and_relation(self, head_qid: str,
    relation_pid: str)
    -> tp.Dict[str, tp.List]:

def get_tail_values_given_head_and_relation(self, head_qid: str,
    relation_pid: str) -> tp.List[str]:
```


E.5 COT AND IO

E.5.1 COT PROMPT

Q: 哪个状态是由乔治·华盛顿大学的乔治·华盛顿殖民者男子篮球队代表的？

A: 首先，该教育机构的体育队名为乔治·华盛顿殖民者男子篮球队，其所在的大学是乔治·华盛顿大学（George Washington University）。其次，乔治·华盛顿大学位于华盛顿特区（Washington D.C.）。答案是华盛顿特区。

Q: 谁将普拉马塔·乔杜里（Pramatha Chaudhuri）列为影响，并撰写了《贾纳·甘娜·曼娜》（Jana Gana Mana）？

A: 首先，巴罗托·巴吉奥·比达塔（Bharoto Bhagyo Bidhata）撰写了《贾纳·甘娜·曼娜》。其次，巴罗托·巴吉奥·比达塔将普拉马塔·乔杜里列为影响。答案是巴罗托·巴吉奥·比达塔。

Q: 哪位艺术家因《你让我疯狂》（You Drive Me Crazy）被提名奖项？

A: 首先，因《你让我疯狂》被提名奖项的艺术家是布兰妮·斯皮尔斯（Britney Spears）。答案是杰森·艾伦·亚历山大（Jason Allen Alexander）。

Q: 哪位出生于齐根（Siegen）的人影响了文森特·梵高（Vincent Van Gogh）的作品？

A: 首先，彼得·保罗·鲁本斯（Peter Paul Rubens）、克劳德·莫奈（Claude Monet）等影响了文森特·梵高的作品。其次，彼得·保罗·鲁本斯出生于齐根。答案是彼得·保罗·鲁本斯。

Q: 哪个接近俄罗斯的国家是米哈伊尔·萨卡什维利（Mikheil Saakashvili）担任政府职位的地方？

A: 首先，中国、挪威、芬兰、爱沙尼亚和格鲁吉亚（Georgia）接近俄罗斯。其次，米哈伊尔·萨卡什维利在格鲁吉亚担任政府职位。答案是格鲁吉亚。

Q: 演员在哪种药物过量后扮演了尿烯轮子家伙（Urethane Wheels Guy）？

A: 首先，米切尔·李·赫德伯格（Mitchell Lee Hedberg）扮演了角色尿烯轮子家伙。其次，米切尔·李·赫德伯格过量服用了海洛因（Heroin）。答案是海洛因。

Q: {查询}

A:

E.5.2 IO PROMPT

Q: 什么状态 (state) 是乔治·华盛顿大学（George Washington University）在体育上由乔治华盛顿殖民者男篮（Colonials men's basketball）代表的？

A: 华盛顿特区 (Washington, D.C.)。

Q: 谁列出了普拉马塔·查乌杜里 (Pramatha Chaudhuri) 的影响，并写了《印度之歌》（Jana Gana Mana）？

A: 巴哈罗托·巴基欧·比达塔 (Bharoto Bhagyo Bidhata)。

Q: 谁是因《你让我疯狂》（You Drive Me Crazy）而获得提名的艺术家？

A: 杰森·亚伦·亚历山大 (Jason Allen Alexander)。

Q: 谁是出生在齐根（Siegen）的，影响了文森特·梵高（Vincent Van Gogh）作品的人？

```
def get_external_id_given_head_and_relation(self, head_qid: str,
    relation_pid: str) -> tp.List[str]:

def mid2qid(self, mid: str) -> str:
```

E.3 ToG

E.3.1 Relation Prune

Please retrieve k relations (separated by semicolon) that contribute to the question and rate their contribution on a scale from 0 to 1 (the sum of the scores of k relations is 1).

`In-Context Few-shot`

Q: {Query}

Topic Entity: {Topic Entity}

Relations: {list of relations}

A:

E.3.2 Entity Prune

Please score the entities' contribution to the question on a scale from 0 to 1 (the sum of the scores of all entities is 1).

`In-Context Few-shot`

Q: {Query}

Relation: {Current Relation}

Entites: {list of entities}

Score:

E.3.3 Reasoning

Given a question and the associated retrieved knowledge graph triples (entity, relation, entity), you are asked to answer whether it's sufficient for you to answer the question with these triples and your knowledge (Yes or No).

`In-Context Few-shot`

Q: {Query}

Knowledge triples: {Explored Paths}

A:

E.3.4 Generate

Given a question and the associated retrieved knowledge graph triples (entity, relation, entity), you are asked to answer the question with these triples and your own knowledge.

A: 彼得·保罗·鲁本斯 (Peter Paul Rubens)。

Q: 哪个靠近俄罗斯的国家是米哈伊尔·萨卡什维利（Mikheil Saakashvili）担任政府职务的地方？

A: 格鲁吉亚 (Georgia)。

Q: 演员在扮演尿烯轮子家伙（Urethane Wheels Guy）时 overdose 的药物是什么？

A: 海洛因 (Heroin)。

Q: {Query}

A:

`In-Context Few-shot`

Q: {Query}

Knowledge triples: {Explored Paths}

A:

E.4 ToG-R

E.4.1 Reasoning

Please answer the question using Topic Entity, Relations Chains and their Candidate Entities that contribute to the question, you are asked to answer whether it's sufficient for you to answer the question with these triples and your knowledge (Yes or No).

`In-Context Few-shot`

Q: {Query}

Topic Entity, with relations chains, and their candidate entities: {Explored Relation Chains}

A:

E.5 CoT and IO

E.5.1 CoT prompt

Q: What state is home to the university that is represented in sports by George Washington Colonials men's basketball?

A: First, the education institution has a sports team named George Washington Colonials men's basketball in is George Washington University , Second, George Washington University is in Washington D.C. The answer is Washington, D.C.

Q: Who lists Pramatha Chaudhuri as an influence and wrote Jana Gana Mana?

A: First, Bharoto Bhagyo Bidhata wrote Jana Gana Mana. Second, Bharoto Bhagyo Bidhata lists Pramatha Chaudhuri as an influence. The answer is Bharoto Bhagyo Bidhata.

Q: Who was the artist nominated for an award for You Drive Me Crazy?

A: First, the artist nominated for an award for You Drive Me Crazy is Britney Spears. The answer is Jason Allen Alexander.

Q: What person born in Siegen influenced the work of Vincent Van Gogh?

A: First, Peter Paul Rubens, Claude Monet and etc. influenced the work of Vincent Van Gogh. Second, Peter Paul Rubens born in Siegen. The answer is Peter Paul Rubens.

Q: What is the country close to Russia where Mikheil Saakashvii holds a government position?

A: First, China, Norway, Finland, Estonia and Georgia is close to Russia. Second, Mikheil Saakashvii holds a government position at Georgia. The answer is Georgia.

Q: What drug did the actor who portrayed the character Urethane Wheels Guy overdosed on?

A: First, Mitchell Lee Hedberg portrayed character Urethane Wheels Guy. Second, Mitchell Lee Hedberg overdose Heroin. The answer is Heroin.

Q: {Query}

A:

E.5.2 IO PROMPT

Q: What state is home to the university that is represented in sports by George Washington Colonials men's basketball?

A: Washington, D.C.

Q: Who lists Pramatha Chaudhuri as an influence and wrote Jana Gana Mana?

A: Bharoto Bhagyo Bidhata.

Q: Who was the artist nominated for an award for You Drive Me Crazy?

A: Jason Allen Alexander.

Q: What person born in Siegen influenced the work of Vincent Van Gogh?

A: Peter Paul Rubens.

Q: What is the country close to Russia where Mikheil Saakashvii holds a government position?

A: Georgia.

Q: What drug did the actor who portrayed the character Urethane Wheels Guy overdosed on?

A: Heroin.

Q: {Query}

A: