**THINK-ON-GRAPH: DEEP AND RESPONSIBLE REASON-ING OF LARGE LANGUAGE MODEL ON KNOWLEDGE GRAPH**

图上思考：在知识图谱上进行深入和负责任的大型语言模型推理

尽管大型语言模型（LLMs）在各种任务中取得了显著的成功，但它们经常在深度和负责任的推理场景中遇到幻觉问题。这些问题可以通过在LLM推理中引入外部知识图谱（KG）来部分解决。在本文中，我们提出了一种新的LLM-KG集成范式“LLM ⊗KG”，将LLM视为一个代理，与KG上的相关实体和关系进行交互性探索，并基于检索到的知识进行推理。我们通过引入一种称为“Think-on-Graph（ToG）”的新方法来进一步实现这一范式，在这种方法中，LLM代理在KG上迭代执行波束搜索，发现最有前景的推理路径，并返回最可能的推理结果。我们进行了一系列精心设计的实验，来检验和阐明ToG的以下优点：1）与LLMs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，具有知识可追溯性和知识可纠正性的能力；3）ToG为不同的LLMs、KGs和提示策略提供了灵活的即插即用框架，无需额外的训练成本；4）在某些情景下，ToG与小型LLM模型的性能可能超过大型LLM（如GPT-4），这降低了LLM部署和应用的成本。作为一种无需训练的方法，具有较低的计算成本和更好的通用性，ToG在9个数据集中的6个数据集中整体实现了最先进技术（SOTA），而大多数先前的SOTA依赖于额外的训练。

1.Introduction

大型语言模型（LLMs）在各种自然语言处理任务中表现出色。这些模型利用预训练技术应用于大量文本语料库，生成连贯且上下文适当的响应。尽管它们有着出色的表现，但在面对需要深入和负责任推理的复杂知识推理任务时，LLMs存在着重大限制。首先，对于需要超出预训练阶段所包含的专业知识的问题（图1a中过时知识），或者对于需要长逻辑链和多跳知识推理的问题，LLMs通常无法提供准确的答案。其次，LLMs缺乏责任、可解释性和透明性，引发了关于幻觉或有毒文本风险的担忧。第三，LLMs的训练过程通常昂贵且耗时，使得保持它们的知识更新成为具有挑战性的任务。

认识到这些挑战，一个自然且有前景的解决方案是将外部知识（如知识图谱（KGs））纳入，以帮助改进LLM的推理能力。知识图谱提供了知识的结构化、明确和可编辑的表示，为减轻LLM的局限性提供了一种互补策略，已经探索了将知识图谱作为外部知识源来减轻LLM中的幻觉问题。这些方法遵循一个常规流程：从知识图谱中检索信息，相应地增加提示，然后将增加的提示馈送到LLMs中（如图1b所示）。在本文中，我们将这种范式称为“LLM⊕KG”。尽管旨在整合LLM和KG的力量，在这种范式中，LLM扮演着将输入问题转换为机器可理解命令以进行KG搜索和推理的翻译器的角色，但它并不直接参与图推理过程。不幸的是，松散耦合的LLM⊕KG范式有其自身的局限性，其成功程度严重依赖于知识图谱的完整性和高质量。例如，在图1b中，尽管LLM成功识别出回答问题所需的必要关系类型，但缺少“多数党”关系导致未能检索到正确答案。

在这些考虑基础上，我们提出了一种新的紧密耦合的“LLM ⊗ KG”范式，在这种范式中，知识图谱和LLMs协同工作，在图推理的每个步骤中相互补充彼此的能力。图1c提供了一个例子，说明了LLM ⊗ KG的优势。在这个例子中，导致图1b失败的缺失关系“多数党”可以由LLM代理发现的具有动态推理能力的参考三元组（澳大利亚，总理，安东尼·阿尔班斯），以及安东尼·阿尔班斯的政党成员身份（来自LLM固有知识）来补充。通过这种方式，LLM成功地从知识图谱中检索到可靠知识，生成了正确答案。作为这种范式的一个实现，我们提出了一种算法框架“Think-on-Graph”（意为：LLMs沿着知识图谱的推理路径“思考”，逐步进行，简称为ToG），用于深入、负责任和高效的LLM推理。在KG/LLM推理中使用波束搜索算法，ToG允许LLM动态地探索多个知识图谱中的推理路径，并相应地做出决策。给定一个输入问题，ToG首先识别初始实体，然后通过探索（通过“在图上”步骤在知识图谱中寻找相关三元组）和推理（通过“思考”步骤决定最相关的三元组）迭代调用LLM从知识图谱中检索相关三元组，直到通过波束搜索中的前N个推理路径收集到足够的信息来回答问题（由LLMs在“思考”步骤中判断），或者达到预定义的最大搜索深度。

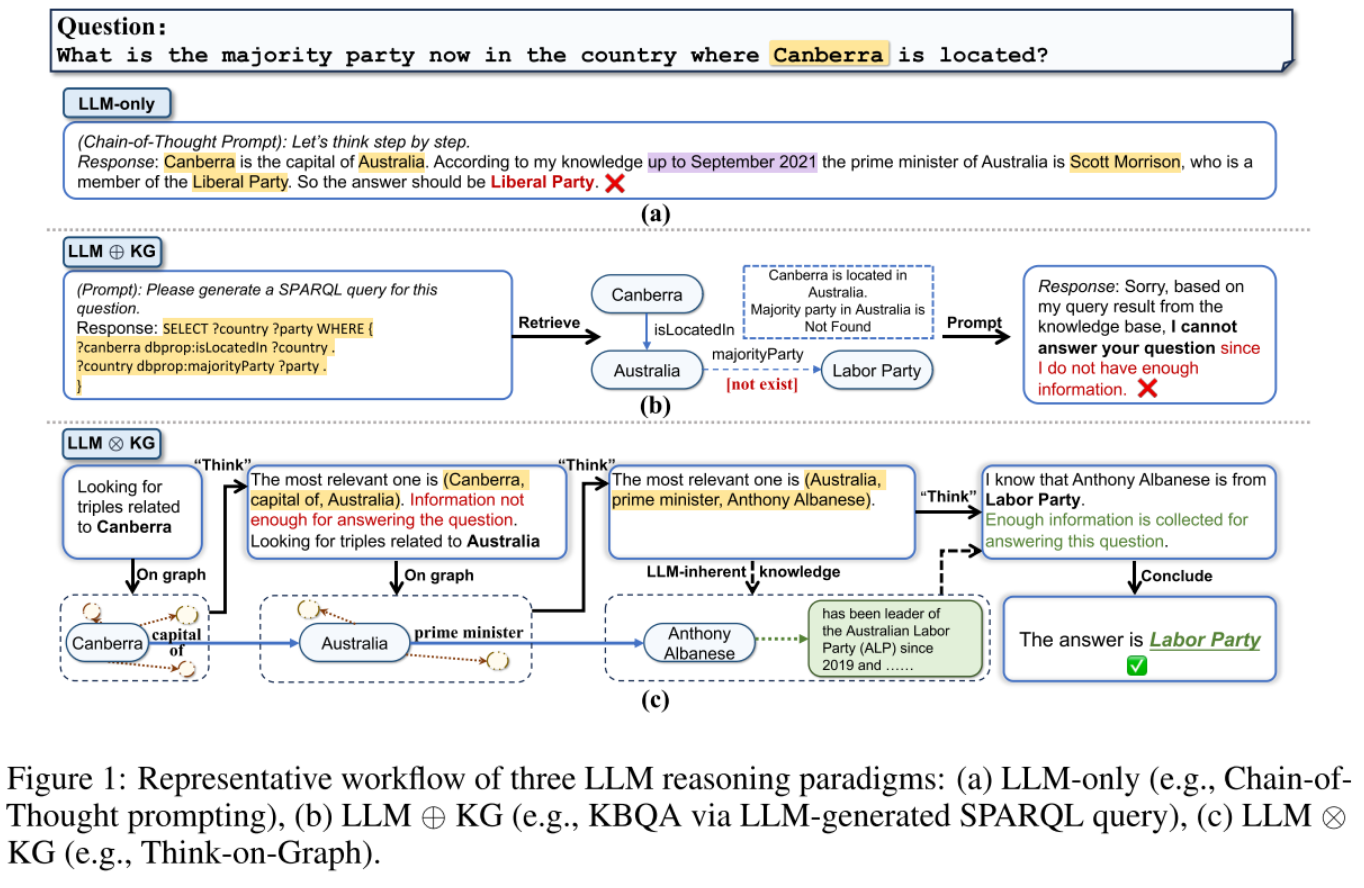


图1：三种LLM推理范式的代表性工作流程：(a) 仅LLM（例如，思维链提示），(b) LLM ⊕ KG（例如，通过LLM生成的SPARQL查询进行知识库问答），(c) LLM ⊗ KG（例如，图上思考）。

{

问:在堪培拉所在的国家，现在哪个政党是多数党?

**LLMonly**

(思维链提示):让我们一步一步地思考。答:堪培拉是澳大利亚的首都。据我所知，截至2021年9月，澳大利亚的总理是斯科特·莫里森，他是自由党成员。所以答案应该是自由党。LLM-only❌

**LLM ⊕ KG**

(提示):请为这个问题生成一个SPARQL查询。

响应：选择 ?country ?party 在 {

?canberra dbprop:isLocatedIn ?country .

?country dbprop:majorityParty ?party .}

这是一个简单的SPARQL查询，用于检索关于国家（country）和其主要政党（majority party）的信息。在这个查询中，通过两个三元组模式来匹配数据图中符合条件的数据，其中第一个三元组模式表示“canberra位于哪个国家”，而第二个三元组模式表示“该国家的主要政党是什么”。

没有找到多数党

Response:很抱歉，根据我在知识库的查询结果，我没有足够的信息，无法回答你的问题。

**LLM ⊗ KG**

寻找与堪培拉相关的三元组

最相关的一个是(堪培拉，澳大利亚首都)。信息不足以回答问题。寻找与澳大利亚相关的三元组

最相关的是(澳大利亚总理，安东尼·艾博年)。

安东尼·艾博年自2019年以来一直是澳大利亚工党(ALP)的领导人，......

我知道安东尼·艾博年是工党的。收集了足够的信息来回答这个问题。

答案是工党

}

ToG的优势可以简化为：(1) 深度推理：ToG从知识图谱中提取多样化和多跳推理路径作为LLM推理的基础，增强了LLMs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LLMs推理过程的可解释性，并使模型输出的来源进行追踪和校正成为可能。 (3) 灵活性和效率：a) ToG是一个即插即用的框架，可以无缝应用于各种LLMs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，而不是通过LLM进行昂贵且缓慢的知识更新。 c) ToG提升了小型LLMs（例如LLAMA2-70B）的推理能力，使其能够与大型LLMs（例如GPT-4）竞争。

2 Methods

ToG通过要求LLM在知识图谱上执行波束搜索来实现“LLM⊗KG”范式。具体来说，它促使LLM迭代地在知识图谱上探索多条可能的推理路径，直到LLM确定根据当前推理路径可以回答问题为止。ToG在每次迭代后不断更新和维护问题x的前N个推理路径P = {p1, p2, . . . , pN}，其中N表示波束搜索的宽度。ToG的整个推理过程包括以下三个阶段：初始化、探索和推理。

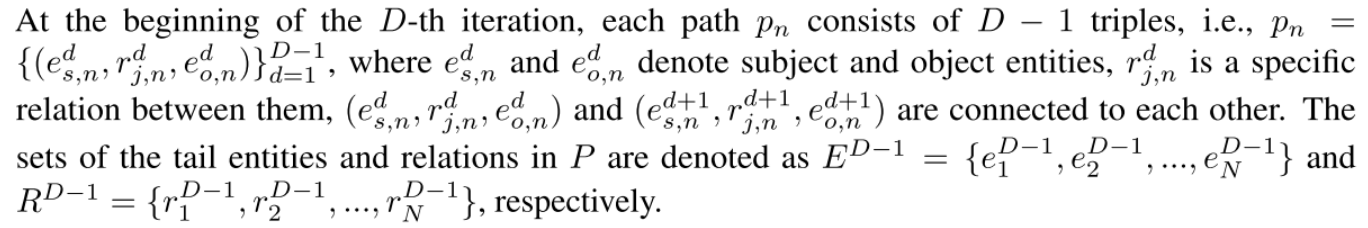
2.1 Think on Graph

2.1.1 Initialization of Graph Search图搜索的初始化

在给定一个问题的情况下，ToG利用基础的LLM来定位在知识图谱上推理路径的初始实体。这个阶段可以被看作是前N个推理路径P的初始化过程。ToG首先促使LLMs自动提取问题中的主题实体，并获取与该问题相关的前N个主题实体，需要注意的是，主题实体的数量可能少于N。

2.1.2 Exploration 探索

在第D次迭代开始时，每个路径pn由D−1个三元组组成，



表示主体实体和客体实体，它们之间的特定关系，互相连接，路径P中尾部实体和关系的集合分别表示为

在第D次迭代中的探索阶段旨在利用LLM识别基于问题x的、与当前前N个实体集合ED−1的邻近实体中最相关的前N个实体ED，，并使用ED扩展前N个推理路径P。为了解决使用LLM处理众多邻近实体的复杂性，我们实施了一个两步探索策略：首先探索重要关系，然后利用选定的关系指导实体的探索。

关系探索：关系探索是一个从ED−1到RD的深度为1、宽度为N的波束搜索过程，整个过程可以分解为两个步骤:search和Prune。大型语言模型作为代理自动完成此过程。

在第d次迭代开始时，关系探索阶段首先为每条推理路径pn查找与尾实体 相连的关系 这些关系被聚合成。在图2中，表示向内或向外链接到Canberra的所有关系的集合。值得注意的是，Search过程可以通过执行附录E.1和E.2中所示的两个简单的预定义形式查询轻松完成，这使得ToG可以很好地适应不同的kg，而无需任何训练成本。

一旦我们获得了候选关系集合和从关系搜索中扩展出的候选推理路径，我们可以利用LLM基于问题x和候选关系从中选择出以尾部关系结尾的新的前N个推理路径P。这里使用的提示可以在附录E.3.1中找到。如图2所示，在第一次迭代中，LLM从与实体Canberra相关联的所有关系中选择出前3个关系{capital of, country, territory}。由于Canberra是唯一的主题实体，因此前3个候选推理路径更新为{(Canberra, capital of), (Canberra, country), (Canberra, territory)}。

实体探索与关系探索类似，实体探索也是大型语言模型从RD到ED的光束搜索过程，包括搜索和修剪两个步骤

一旦我们从关系探索中获得了新的top-N推理路径P和新的尾关系集RD，对于每个关系路径pn∈P，我们可以通过查询来探索候选实体集，其中，表示pn的尾实体及其关系。我们可以聚合扩展为，并加上尾部实体将top-N推理路径P扩展为。对于所示的例子，可以表示为{Australia, Australia, Australian Capital Territory}。

由于每个候选集合中的实体都是用自然语言表示的，我们可以利用大型语言模型来选择新的top-N推理路径P, P以从取出的尾实体ED结尾。这里使用的提示符可以在附录E.3.2中找到。如图2所示，由于关系Capital of, country和Territory分别只关联到一个尾实体，所以Australia和Australian Capital Territory得分为1，当前推理路径p更新为{(Canberra, Capital of, Australia)， (Canberra, country, Australia)， (Canberra, Territory，Australian Capital Territory)}。

在执行上述两次探索之后，我们重建新的top-N推理路径P，其中每条路径的长度增加1。每个修剪步骤最多需要N个大型语言模型调用。

2.1.3 Reasoning

通过探索过程获得当前的推理路径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使用以查询为输入的推理路径生成答案，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，如果评估结果为负面，我们重复探索和推理步骤，直到评估结果为积极或达到最大搜索深度Dmax。如果算法尚未结束，则表明即使达到了Dmax，ToG仍无法探索推理路径来解决问题。在这种情况下，ToG仅基于LLM中的固有知识生成答案。整个ToG的推理过程包括D个探索阶段、D个评估步骤以及一个生成步骤，最多需要2ND + D + 1次LLM调用。

2.2 RELATION-BASED THINK-ON-GRAPH 基于关系的图上思维

以前的基于语义解析的知识库问答方法主要依赖于问题中的关系信息来生成正式查询受此启发，我们提出了基于关系的ToG（ToG-R），它探索以主题实体为起点的前N个关系链，而不是基于三元组的推理路径。oG-R在每次迭代中依次执行关系搜索、关系剪枝和实体搜索，这与ToG相同。然后，ToG-R根据实体搜索获得的以结尾的所有候选推理路径执行推理步骤。如果LLM确定检索到的候选推理路径不包含足够的信息供LLM回答问题，我们将从候选实体中随机抽样N个实体，并继续到下一次迭代。假设每个实体集合中的实体可能属于相同的实体类并具有相似的邻近关系，对实体集合进行修剪的结果可能对接下来的关系探索影响不大。因此，我们在实体剪枝中使用随机波束搜索而不是LLM受限波束搜索，称为随机剪枝。算法1和算法2展示了ToG和ToG-R的实现细节。ToG-R最多需要ND + D + 1次LLM调用。

与ToG相比，ToG- r提供了两个关键优势:1)它不需要使用大型语言模型来修剪实体的过程，从而降低了总体成本和推理时间。2) ToG-R主要强调关系的文字信息，降低了大型语言模型对中间实体的文字信息缺失或不熟悉时产生错误推理的风险。

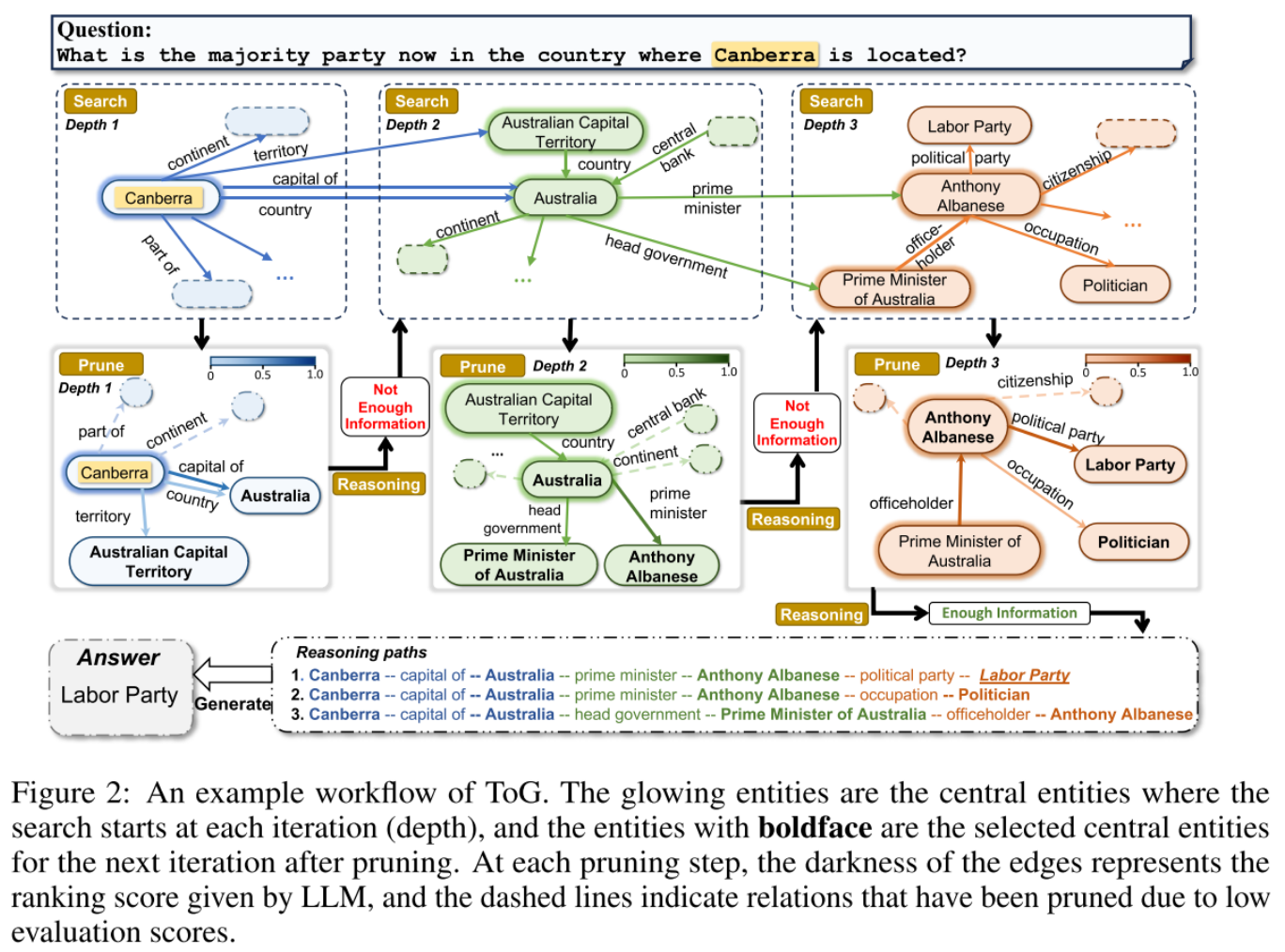


图2:ToG的一个示例工作流。发光的实体是每次迭代(深度)开始搜索的中心实体，黑体字的实体是剪接后下一次迭代选择的中心实体。在每个修剪步骤中，边缘的暗度表示大型语言模型给出的排序分数，虚线表示由于评价分数低而被修剪的关系。

{

问:在堪培拉所在的国家，现在哪个政党是多数党?

}

3 EXPERIMENTS

3.1 EXPERIMENTAL DESIGN

3.1.1 DATASETS AND EVALUATION METRICS 数据集和评估指标

为了测试ToG在多跳知识密集型推理任务上的能力，我们在五个知识库问答数据集上对ToG进行评估（4个多跳和1个单跳）：CWQ、WebQSP、GrailQA、QALD10-en、Simple Questions。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；两个槽填充数据集：T-REx（和Zero-Shot RE；以及一个事实核查数据集：Creak。值得注意的是，对于两个大型数据集GrailQA和Simple Questions，我们仅随机选择了每个1,000个样本进行测试，以节省计算成本。对于所有数据集，我们使用精确匹配准确率（Hits@1）作为我们的评估指标，遵循之前的研究。

3.1.2 METHODS SELECTED FOR COMPARISON 选择比较的方法

我们将ToG与标准提示（IO提示）、思维链提示（CoT提示）和自洽性（进行比较，这些方法使用了6个上下文示例和“逐步”推理链。此外，对于每个数据集，我们选择了以往的最先进（SOTA）作品进行比较。我们注意到，专门在评估数据集上进行微调的方法通常天生具有优势，相对于没有经过训练的提示方法，但会牺牲在其他数据上的灵活性和泛化能力。因此，为了公平竞争，我们分别与所有基于提示的方法和所有方法中以前的SOTA进行比较。需要注意的是，Tan等人（2023）的论文未参与比较，因为其结果不是基于标准的精确匹配，因此无法进行比较。

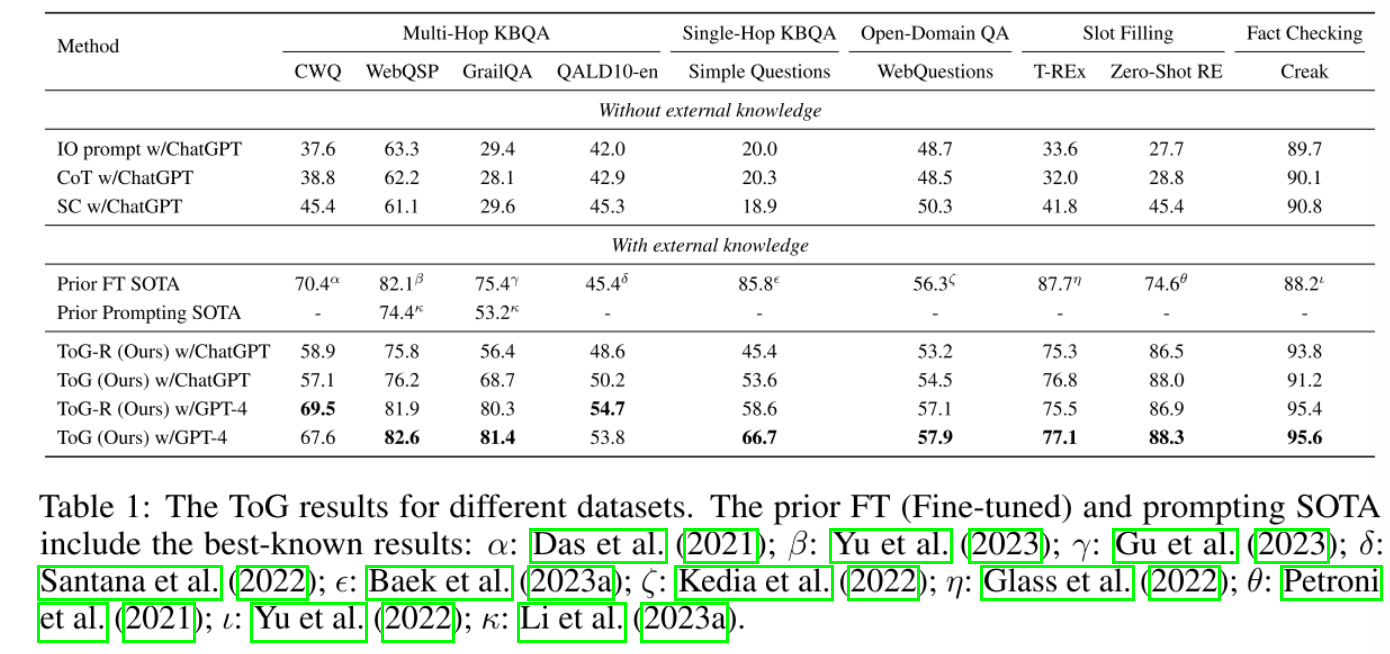


表1:不同数据集的ToG结果。先前的FT(微调)和提示SOTA包括最著名的结果:α: Das等人(2021);β: Yu et al. (2023);γ: Gu等人(2023);δ: Santana et al. (2022);λ: Baek et al. (2023a);ζ: Kedia et al. (2022);η: Glass et al. (2022);θ: Petroni et al. (2021);ι: Yu et al. (2022);κ: Li et al. (2023a)。

3.1.3 EXPERIMENT DETAILS 实验细节

考虑到ToG即插即用的便利性，我们在实验中尝试了三种大型语言模型: ChatGPT, GPT-4和Llama-2。我们使用OpenAI API调用ChatGPT (GPT-3.5-turbo)和GPT-41。Llama2-70B-Chat (Touvron et al.， 2023)使用8个A100-40G不进行量化，其中勘探过程温度参数设置为0.4(增加多样性)，推理过程温度参数设置为0(保证再现性)。生成的最大令牌长度设置为256。在所有实验中，我们都将波束搜索的宽度N和深度Dmax设置为3。CWQ、WebQSP、GrailQA、Simple Questions和Webquestions使用Freebase (Bollacker et al.， 2008)作为KG, QALD10-en、T-REx、零样本学习 RE和Creak使用Wikidata (vande ci´c & Krötzsch, 2014)作为KG。对于所有数据集，我们使用了5次tog推理提示。

3.2 MAIN RESULTS

3.2.1 COMPARISON TO OTHER METHODS

由于CoT使用外部KG来增强大型语言模型，我们首先将其与那些利用外部知识的方法进行比较。从图1中我们可以看到，即使ToG是一种不需要训练的基于提示的方法，与那些用评估数据训练的微调方法相比，ToG具有天然的劣势，但在9个数据集中，使用GPT-4的ToG仍然在6个数据集中达到了新的SOTA性能，包括WebQSP、GrailQA、QALD10-en, WebQuestions，零样本学习RE和Creak。即使对于一些没有SOTA的数据集，例如CWQ, CoT的性能也已经接近SOTA (69.5% vs . 70.4%)。与所有基于提升的方法相比，GPT-4的ToG及其弱版本的ChatGPT ToG都能在所有数据集的竞争中获胜。特别是，在开放域QA数据集WebQuestions上1.6%的改进证明了ToG在开放域QA任务上的通用性。我们还注意到ToG在单跳KBQA数据集上的性能不如它在其他数据集上的性能。这些结果表明，ToG在一般情况下对多跳数据集更有效，这支持了我们关于ToG增强大型语言模型深度推理能力的观点。

从图1中我们还可以看到，与那些没有利用外部知识的方法(如IO、CoT和SC提示方法)相比，ToG的优势更为显著。例如，在GrailQA和零样本学习RE上，性能分别提高了51.8%和42.9%。事实证明，外部KG的收益在推理中是不可忽视的。在大多数数据集上，ToG优于ToG- r，因为与ToG- r检索的关系链相比，基于三重的推理路径提供了额外的中间实体信息。对ToG生成的答案的更详细分析可以在附录B.2中查看。为了便于比较，在附录C中报告了之前方法对每个数据集的处理结果。

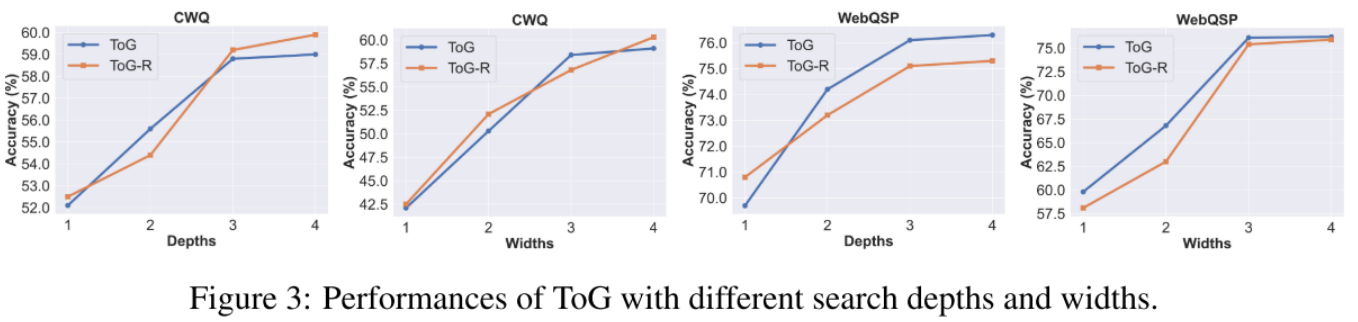
3.2.2 PERFORMANCES WITH DIFFERENT BACKBONE MODELS 不同骨干模型的性能

考虑到ToG的即插即用灵活性，我们评估了不同的骨干模型如何影响其在CWQ和WebQSP两个数据集上的性能。表2显示，正如我们所料，CoT的性能随着主干模型(GPT-4 > ChatGPT > Llama-2)的大小(也部分反映了推理能力)而提高。此外，我们看到，主干模型越大，CoT与ToG之间的差距越大(在CWQ上，Llama-2的增益从18.5%增加到23.5%，在WebQSP上，Llama-2的增益从11.5%增加到15.3%)，这表明使用更强大的大型语言模型可以挖掘更多的KG潜力。此外，即使使用最小型号的lama-2 (70B参数)，ToG也优于使用GPT-4的CoT。这意味着大型语言模型部署和应用的技术路线要便宜得多，也就是说，具有廉价小型大型语言模型的TOG可能是取代昂贵大型大型语言模型的候选方案，特别是在外部kg可以覆盖的垂直场景中。

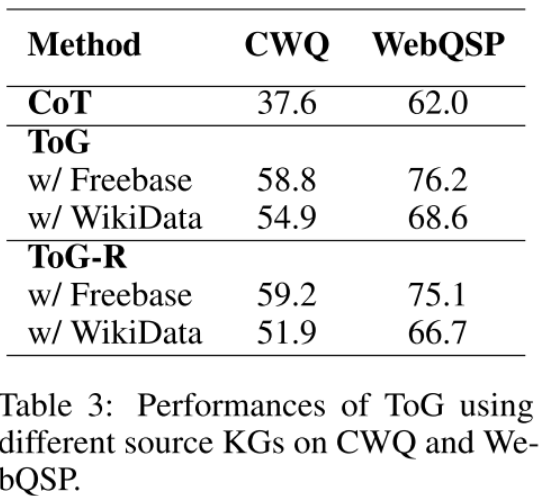
3.2.3 ABLATION STUDY

我们进行了各种消融研究，以了解不同因素在ToG中的重要性。我们在CWQ和WebQSP测试集的两个子集上进行消融研究，每个子集包含1000个随机抽样问题。

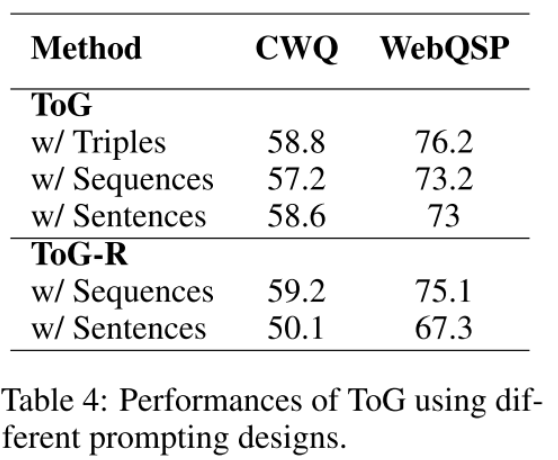
搜索深度和宽度对ToG有影响吗?为了探索搜索深度Dmax和波束宽度N对ToG性能的影响，我们在深度1 ~ 4和宽度1 ~ 4的设置下进行了实验。如图3所示，ToG的性能随着搜索深度和宽度的增加而提高。这也意味着随着勘探深度和广度的增加，ToG的性能有可能得到改善。然而，考虑到计算成本(随着深度线性增加)，我们将深度和宽度都设置为3作为默认的实验设置。另一方面，当深度超过3时，性能增长减弱。这主要是因为只有一小部分问题的推理深度(基于SPARQL中的关系数量，如附录中的图12所示)大于3。



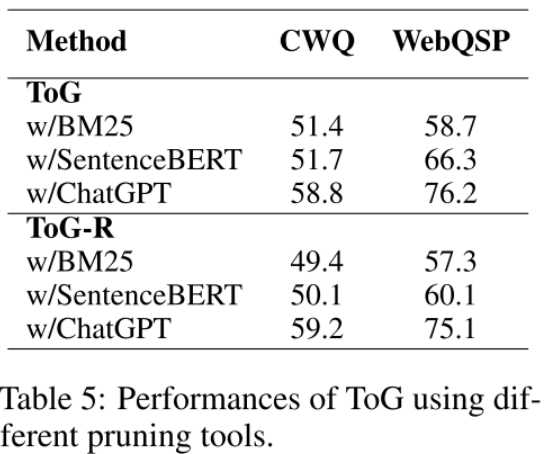
不同的kg会影响ToG的表现吗?ToG的主要优点之一是它的即插即用功能。如表3所示，与CoT相比，ToG在CWQ和WebQSP上使用不同的源KGs取得了显著的改进。另一方面，不同的源KGs对ToG性能的影响可能不同。值得注意的是，Freebase在CWQ和WebQSP上带来了比Wikidata更显著的改进，因为这两个数据集都是在Freebase上构建的。此外，在像Wikidata这样的非常大的KG中，搜索和修剪过程相对具有挑战性。



不同的提示设计对《ToG》有何影响?我们进行了额外的实验，以确定哪种类型的提示表示可以很好地适用于我们的方法。结果如表4所示。"Triples"表示使用三重格式作为提示符来表示多个路径，例如"(Canberra, capital of, Australia)， (Australia, prime minister, Anthony Albanese)"。“序列”指的是序列格式的使用，如图2所示。“句子”涉及将三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”。结果表明，基于三元表示的推理路径的使用产生了最高程度的效率和优越的性能。相反，在考虑ToG-R时，每个推理路径都是从主题实体开始的关系链，这使得它与基于三重的提示表示不兼容。因此，将ToG-R转换为自然语言形式会产生过长的提示，从而导致性能显著下降。



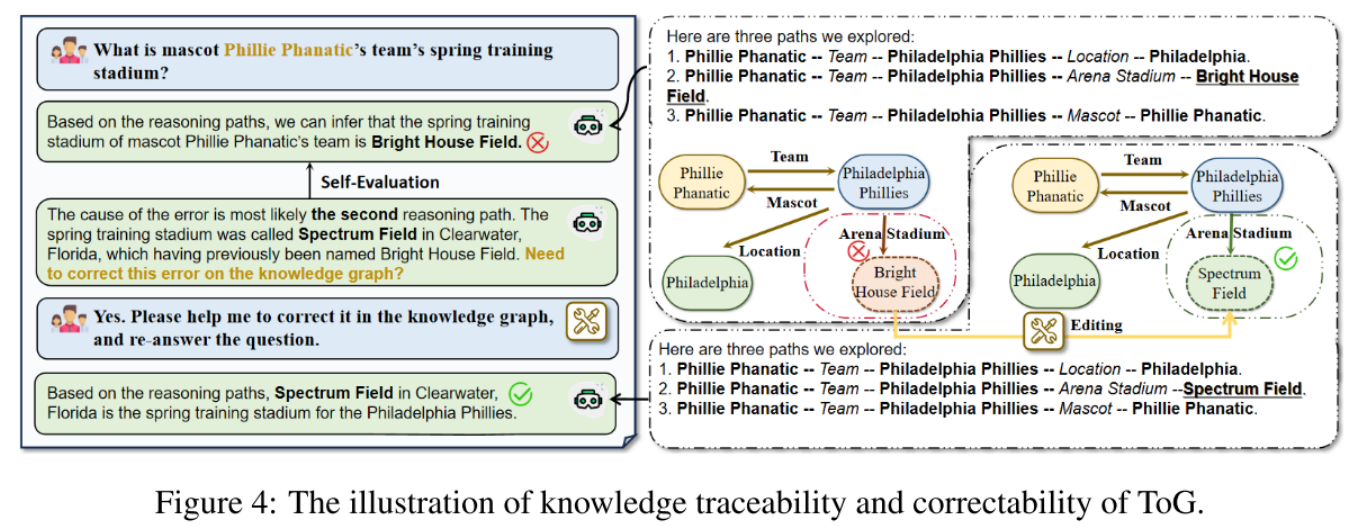
比较不同修剪工具的影响。除了大型语言模型之外，可以测量文本相似度的轻量级模型(如BM25和SentenceBERT)可以用作探索阶段的修剪工具。我们可以根据实体和关系与问题的字面相似度选择top-N的实体和关系。我们研究了不同修剪工具对ToG性能的影响，如表5所示。用BM25或SentenceBERT替换大型语言模型会导致我们的方法的性能显著下降。具体来说，CWQ的结果平均下降了8.4%，WebQSP的结果平均下降了15.1%。结果表明，就有效性而言，大型语言模型作为修剪工具表现最佳。另一方面，在使用BM25或SentenceBERT之后，我们只需要D + 1个对大型语言模型的调用，而不是我们在2.1.3节中讨论的第2个+D + 1，这提高了ToG的效率。



我们对种子样本数量的影响以及ToG和朴素束搜索对KG的差异进行了额外的消融研究，详见附录B.1。

3.3 KNOWLEDGE TRACEABILITY AND CORRECTABILITY IN TOG知识的可追溯性可纠正性

KG的质量对于ToG的正确推理非常重要。ToG在大型语言模型推理过程中的一个有趣特性是知识可追溯性和知识可纠错性，它提供了一种利用ToG本身来提高KG质量，降低KG构建和纠错成本的方法。如图4所示，tog的显式推理路径可以显示给用户。如果人类用户/专家或其他大型语言模型发现了ToG答案中的潜在错误或不确定性，那么ToG有能力追溯和检查推理路径，发现可疑的错误三元组，并纠正它们。



以图4中的情况为例。给定输入问题“吉祥物菲利·法纳蒂奇队的春季训练场馆是什么?”， ToG在第一轮中输出错误答案“Bright House Field”。然后ToG追溯了所有的推理路径，定位了错误的原因可能来自于第二条推理路径(Phillie Phanatic Team−−−→Philadelphia Phillies Arena Stadium−−−−−−−→Bright House Field)，并分析了错误来自于过时的三人组(Philadelphia Phillies，竞技场体育场(Arena Stadium, Bright House Field)。根据ToG的提示，用户可以要求大型语言模型纠正这个错误，用正确的信息回答相同的问题。这个例子表明，ToG不仅通过KG增强了大型语言模型，而且通过大型语言模型提高了KG的质量，称为知识注入(Moiseev et al.， 2022)。

5 CONCLUSION

我们引入了大型语言模型⊗KG范式，以紧密耦合的方式集成大型语言模型和KG，并提出了图上思考(Think-on-Graph, ToG)算法框架，该框架利用大型语言模型作为参与KG推理的代理，以做出更好的决策。实验结果表明，ToG在不增加训练成本的情况下优于现有的基于微调和基于提示的方法，并减轻了大型语言模型的幻觉问题。