第5章

大模型推理能力优化

随着大模型的发展，推理能力的优化成为当前研究的一个重要方向。大模型不仅能通过庞大的参数和数据学习大量知识，还能在多种任务中表现出强大的推理能力。然而，这些模型也面临着推理过程中的效率、精度、可解释性等方面的挑战。本章将深入探讨大模型推理能力的提升方法，介绍不同的优化策略，包括基于提示工程、数据集构建、蒙特卡洛方法等手段。同时，还将讨论知识编辑技术及其在大模型优化中的作用，帮助读者更好地理解和应用大模型。

本章学习目标：

- 掌握大模型推理的基本任务。
- 学习不同的推理能力提升方法，如基于提示工程的方法、基于数据集构建的方法、基于蒙特卡洛的方法。
- 了解逆转诅咒的缓解方法，如重排训练、演绎闭合训练等方法。
- 掌握知识编辑的基本概念与方法。
- 理解知识编辑方法的评估方法。

5.1

5.1 大模型推理能力

5.1.1 大模型推理的主要任务

为了更深入地了解大型语言模型的推理能力，了解相关的推理任务至关重要，这些任务被广泛认为需要模型的推理能力才能有效解决。目前推理任务主要集中于文本大模型，在本节中，将介绍 3 种不同类型的纯语言推理任务：算术推理、常识性推理、符号推理。

1. 算术推理

解决数学问题通常需要一步或多步算术推理。基于对输入问题、隐含算术运算和概念知识的理解，解题者需要根据已有知识和已知条件推导出一系列结果，从而得到最终答案。例如，GSM8K[161]、SVAMP[160]、ASDiv[645]和 MAWPS[646]的基准测试主要包含小学水平的数学知识，即加、减、乘、除等基本运算。MathQA[167]和 AQuA[647]包括来自一些考试的基准测试。MATH[164]以极具挑战性的数学问题为特色，包括代数、计算、几何、组合数学、数论等，该基准测试要求求解者拥有大量高级数学知识和数学推理技巧以及多步思考的能力。

2. 常识性推理

常识性推理指的是基于日常生活经验和世界常识进行的推理能力。它涉及理解隐含的知识、处理模糊信息,以及做出符合现实世界逻辑的判断。常识性推理具有隐含性、普适性、模糊性、动态性等特点。常识性推理是人类智能的重要组成部分,能够帮助人们在面对不确定或不完全信息时做出合理的推断。经典的常识推理的数据集,例如 HellaSwag[132]、Winogrande[648]、Socialiqa[135],这些数据集通常不需要多步骤的推理过程,但是常识涉及大量的细节,并且常识推理往往模棱两可,AI 需要处理模糊和非结构化的信息。

3. 符号推理

符号推理是一种基于符号表示和逻辑规则的推理方式,通常用于精确地解决问题,进行数学推导、逻辑推理和知识表示。符号推理的核心在于使用明确的规则、定义和关系,通过符号(变量、常量、运算符等)进行逻辑推断。符号推理具有精确性、可解释性、基于规则、离散性,例如,如果 A 为真,且 A→B 为真,则 B 为真。以 PrOntoQA[649]、SimpleLogic[650]、FOLIO[154]、ProofWriter[651]等数据集为例,在这些任务中,在上下文中给出了一组事实和逻辑规则,并且需要模型根据逻辑运算来证明公式。虽然语言模型展示了理解简单符号操作的能力,但它们被认为不太擅长复杂的符号推理任务。

5.1.2 大模型推理能力分析

在深度学习进步和 Web 规模数据集推出的推动下,大型语言模型已成为迈向通用人工智能(Artificial General Intelligence,AGI)的变革性范式。这些大规模 AI 模型通常采用 Transformer 架构,并通过下一个标记预测任务在大规模文本语料库上进行预训练。神经缩放定律表明,随着模型大小和训练数据的增加,它们的性能会显著提高。更重要的是,LLM 还可以解锁小模型中所没有的能力,例如上下文学习[652]、角色扮演和类比推理。这些功能使 LLM 超越了自然语言处理问题,以促进更广泛的任务,例如代码生成[654]、机器人控制[6]和 autonomous agent[655]。现阶段 LLM 推理能力主要存在两大主要问题:思维偏差和逆转诅咒。思维偏差主要源于知识的结构和训练数据结构的相似度影响,而逆转诅咒主要源于单向的事实和 next-token 的自回归训练方法影响。此外,还有研究[656]表明常识问题的稳定性可能并不乐观。

1. 大模型的思维偏差

研究表明[657],事实知识的适当结构对于 LLM 在下游任务上的成功至关重要。遵循特定结构的训练数据使模型能够在提供足够的潜在答案(例如,可用选项)时提供正确的答案。然而,当训练文档偏离模型的首选结构时,它们的知识应用能力可能会变得不稳定,甚至违反直觉。在实验中,模型的输出较为随机,并且简单延长训练时间也并不能打破这种思维偏差。

2. 大模型的逆转诅咒

逆转诅咒依然是大模型推理中难以解决的问题。逆转诅咒是指在 A is B 形式的文档上训练的 LLM 无法推广到反向版本 B is A,这一问题包含了两个方面:一个方面是 A=B 则 A→B 且 B→A;另一方面,<entity> is <description>不能反向推广,而且即使是最强大的 LLM 也难以逆转它们学习到的事实。研究表明逆转诅咒是(有效)模型权重不对称的结果,即在训练过程中从标记 A 到标记 B 的权重增加并不一定会导致权重从 B 到 A 的增

加。Lv 等[658]表明逆转诅咒可能部分归因于 next-token 预测的训练目标，next-token 的训练方式不能很好地预测先行词。Zhu 等[659]对单层 Transformer 进行了理论分析，表明完成任务的逆转诅咒源于梯度下降的训练方法。逆转诅咒揭示了大模型存在的泛化问题，传统知识图谱遵循着恒等的对称性，而逆转诅咒揭示了当前 LLM 在逻辑推理方面泛化存在的问题，目前认为逆转诅咒和偏差的主要来源是基于自回归梯度下降的训练方法和数据集文档结构的共同作用，自回归的梯度下降更新是短视的，并且取决于 A 对 B 的逻辑结构，并不要求从 B 预测 A[660]。

3. 常识问题的稳定性

研究表明[656]，针对使用常识性推理，即使是 10 岁的儿童也能轻松解决，但稍有设计，例如形式如下的问题：Alice 有 N 个兄弟，她还有 M 个姐妹，问 Alice 的兄弟有几个姐妹？此类问题即使是最先进的模型也很难得到准确的回答。其不仅受到数值计算的影响，而且 N 和 M 的微小变化可能导致模型性能的强烈波动。这表明即使对于简单的问题，仍然需要额外的方式来帮助 LLM 优化输出。

4. 数值问题上的误差

大型语言模型在处理简单的数值问题（例如比较两个小数字）时经常出错。研究表明[653]，LLM 在处理数值问题时，错误的预测在字符串空间中接近正确答案，但在数值空间中不接近，即大多数误差是 10、100 等的精确倍数，表明 LLM 在算术问题上展示的分散误差源于数字的碎片化数字表示，而并非线性的数学表示。

5.1.3　基于提示工程的方法

在提示工程方面，先人研究主要是通过构建中间步骤赋予模型推理能力的方法，将多步骤问题分解为多个中间步骤，提高模型推理的准确性，并使得推理过程更加透明。研究表明，延长思维链（CoT）中的推理步骤可以大大提高 LLM 在多个任务中的推理能力[661]。具有更多推理步骤的链，在多步骤推理任务上取得了更好的性能[655]。

然而，CoT 大大提高了推理准确性的同时，有几个关键问题限制了它的实用性：分别是思维链的有效性、模型的有效上下文长度，以及手动编写思维链提示需要大量的人力。CoT 提示的有效性在很大程度上取决于用于提示的少数样本示例的质量和相关性[662]。手动制作这些示例非常耗时，需要大量专业知识，从而限制了 CoT 提示在新领域和新任务中的可扩展性和适用性。前人的研究主要集中在拓展输入结构、压缩输入长度、自动化提示。

1. 问题分析

问题分析是一个关键的初始化过程，是指模型在解决问题之前对问题进行重新构建和分析的过程[663]。尽管 ChatGPT 等 LLM 在理解上下文和生成响应方面表现出色，但在面对含糊查询时难以主动提问澄清问题或拒绝不合理请求，缺乏主动性。为提升这一能力，Deng 等[664]提出了主动思维链（ProCoT）提示策略，通过引导模型进行推理和计划，增强对话系统的主动性。实验结果表明，ProCoT 有效改善了 LLM 在澄清问题、平滑话题的转移和策略性决策方面的不足。

在开放域问答（Open-Domain Question Answering，ODQA）中，用户提出的问题往往存在歧义，这给准确回答带来了很大挑战。传统方法通常采用生成消歧义问题（Disambiguated

Questions,DQ)并针对所有可能解释分别作答的方式来解决这一问题。然而,这种方法在真实应用场景中并不理想,尤其是在语音交互或小屏幕设备上,用户难以接受列出所有可能答案的形式。

为了解决这一问题,Lee 等提出了一种基于澄清问题(Clarification Questions,CQ)的新方法[663]。具体而言,当系统检测到用户提出的问句存在歧义时,不是直接列出所有答案,而是通过提出一个澄清问题,引导用户选择最符合其意图的解释。用户的回答将帮助系统明确用户的需求,从而提供更准确的答案。实验表明,这种 CQ 方法相比于传统的 DQ 方法更受用户欢迎,特别适合真实世界的问答场景。

基于此,设计了一套完整的三阶段处理流程:首先是歧义检测,用于识别用户提出的问题是否存在歧义;其次是澄清问题生成,为检测到的歧义问题生成合适的澄清问题;最后是基于澄清的问答,根据用户对澄清问题的反馈,系统给出精确答案。

2. 复杂任务分解

1) 思维链

思维链是一种将问题分解为中间步骤并在给出最终答案之前解决每个步骤的方法。思维链方法的目标是赋予语言模型生成类似思维链的能力,即一系列连贯的中间推理步骤,从而得出问题的最终答案。思维链的基本原理[63]如图 5.1 所示。

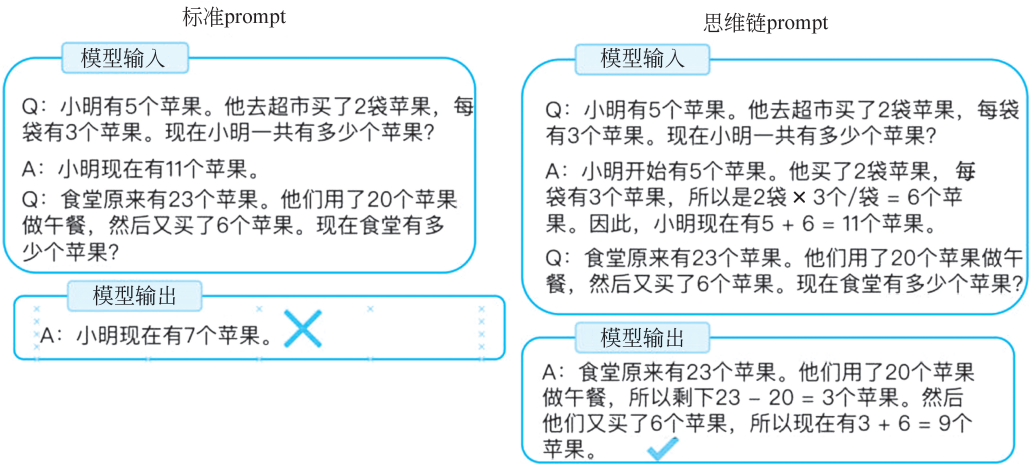

图 5.1 思维链的基本原理

CoT 的特性如下:首先,CoT 原则上允许模型将多步骤问题分解为中间步骤,这意味着可以将额外的计算分配给需要更多推理步骤的问题。其次,思维链为模型的行为提供了一个可解释的窗口,表明它如何得出特定答案,并提供调试推理路径出错的地方的机会。再次,思维链推理可用于数学问题、常识推理和符号操作等任务,并且可能适用于人类可以通过语言解决的任何任务。最后,只需将思维链序列的示例包含在少数镜头提示的示例中,就可以在足够大的现有 LLM 中轻松引出思维链推理[63]。

Feng 等的研究[665]进一步证明了自回归 Transformer 架构下 CoT 要比 RNN 架构有更好的性能,在理论上有足够长的思维链时可以解决 P 完全问题,并且模型确实在一定程度上学习了解决方案,而不是记住数据分布。

CoT 对数据质量有稳健性。Pfau 等研究[666]将 CoT 中间替换为隐藏字符或填充字符

(如“…”),Feng 等[665]对 30%的数据缺少中间 CoT 步骤并涉及单标记损坏的数据进行实验,模型仍然可以达到很高的准确率。

2) 自洽思维链

自洽思维链 CoT-SC[64]在思维链的基础上先从大模型生成一定数量的推理路径,通过边缘化采样的推理路径来确定最佳答案,以在最终答案集中找到最一致的答案。这种方法类似于人类的经验,如果多种不同的思维方式导致相同的答案,那么人们就会更有信心最终答案是正确的。自洽性避免了贪婪解码的重复性和局部最优性,同时减轻了单个采样生成的随机性。通过生成多个推理路径并将它们合并以形成最终答案。与依赖于单个 Chain of thought-trust 相比,自洽涉及生成多个同一查询的独立思维链,允许探索各种推理路径的模型,然后将这些多个输出聚合在一起,通常通过多数投票或其他共识机制,得出最可靠的最终答案。这种方法利用模型输出的多样性来减少错误并提高整体性能。

3) 对比思维链

Chia 等[667]研究分析了各种无效推理类型,发现结合正面和负面的推理通常会提高思维链的有效性。对比思维链被提出来增强语言模型推理。与传统的思维链提示相比,对比思维链的方法对比了有效和无效的答案解释,引导模型生成更准确的推理链,从而提高复杂任务的性能。

4) 思维树

在思维链(CoT)基础上,思维树(ToT)[392]被提出,ToT 将问题构建为树的搜索,其中每个节点作为一个状态,然后对树进行 BFS 或 DFS,ToT 是更一般性的 CoT,ToT 对于问题具有通用性,且不需要额外训练。在创意写作、24 点游戏等方面比 CoT 具有更高的准确性。ToT 允许 LLM 探索多条推理路径,评估潜在结果,迭代选择最有希望的路径。此方法引入了两个关键策略——产生和评估想法:使用对不同的想法进行采样 Chain-of-Thought 提示并提出针对问题约束条件量身定制的顺序思考[667]。

5) 思维图

在 CoT 和 ToT 基础上,为了解决思维过程中可能遇到的更强大表示,比如递归思维的使用,思维图(GoT)[393]被提出,思维图将推理过程建模为有向图,由提示器、解析器、评分验证、控制器等组件构成,并允许用户对一些边进行删减。GoT 将它们生成的信息表示为图表。相比 CoT、ToT,这种方法允许更灵活、更复杂的推理形式在 GoT 中,每个信息或“思想”作为一个顶点,想法之间的关系表示为边。这种基于图的结构促进了思想的组合、完善和生成,使模型可以更有效地处理复杂的多维推理任务。

3. 备选提案

Wang 等[63]生成多个 CoT 并将其与投票机制进行集成,此扩展和类似的扩展不会同时使用多个推理链,因此模型无法在单个推理步骤中访问不同的可能推理链。对此,一些研究人员对备选提案进行了研究。

1) 发散思维链

受到发散思维和收敛思维心理学理论的启发,解决问题涉及两个不同的阶段:发散思维,产生和探索许多想法;然后是收敛思维,涉及考虑这些不同的想法以得出单一的解决方案或响应,发散思维链(DCoT)被提出[668],在域内任务和无形的任务上是有益的,在 CoT 有害的任务上是健壮的。它的性能优于具有自洽性扩展的 CoT,后生成的 CoT 可以自我纠正

先生成的 CoT,而无须任何外部反馈或及时优化。

2) 思想交流

思想交流(EoT),可以在解决问题时实现跨模型通信[669]。LLM 推理往往受到内在理解的制约,缺乏外部洞察力。为了解决这个问题,EoT 从网络拓扑中汲取灵感,集成了 4 种独特的通信范式:内存、报告、中继和辩论。EoT 以具有成本效益的方式实现了这些卓越的结果,这标志着高效和协作解决 AI 问题的巨大进步。

它通过跨模型通信为模型提供了外部见解。开发了 4 种通信范式,并对通信量和信息传播速度进行了全面分析。为了防止错误推理过程的中断,设计了一个置信度评估机制。在数学、常识和符号推理任务,EoT 在性能上有所提高,同时也提供了成本优势。进一步分析表明,EoT 对各种模型具有适应性,更多样化的模型的参与可以进一步增强 EoT 的性能。

4. 自我评估

通过显式反馈(如"Let's check"或"Let's test")来表达自我评价。这种评估能力可以通过两种主要方法得到增强:实施详细的评估标准以灌输自我评估能力,或者利用自我辩论进行交叉验证。

多智能体辩论[670]利用多模型生成与辩论多个模型独立回答问题。将其他模型的答案反馈给每个模型,促使其更新答案。重复多轮此过程,逐步达成一致。也可以灵活控制辩论深度,通过调整提示词引导模型"固执"或"开放",影响辩论轮次和答案质量。可与零样本链式思维(Zero-Shot CoT)等技术结合,进一步提升效果。多轮辩论能纠正模型最初的错误答案,提升一致性与准确性。

Corex 框架[671]引入了 3 种协作模式——讨论、审查和检索,以增强不同代理之间的推理能力。讨论是指代理以小组形式工作,通过迭代讨论来完善推理和预测,从而获得不同的见解并减少错误。审查是指代理审查彼此的推理链和代码以确保正确性并提出改进建议。检索是指一个代理评估来自其他代理的推理链,根据对正确答案的忠实度对其进行评分,以选择最可靠的响应。Corex 框架在实现高精度的同时很大限度减少了推理开销。

5. 自我纠正

在 LLM 推理过程中,自我纠正(SC)是指通过反映先前响应中的错误来迭代改进未来响应的能力。大型语言模型中的自我纠正特征通常可以通过两种方法实现:内源性 SC 和外源性 SC[672]。

1) 外源性自我纠正

外源性自我纠正是指大型语言模型使用外部知识来源来提炼自己的答案。此过程使模型能够根据更新或专业信息进行交叉引用和纠正潜在错误,从而提高其响应的准确性和可靠性。由于深度学习模型在生成模型流行之前主要用于分类任务,因此重点是知识蒸馏,其中更强大的 DL 模型在较小模型的训练阶段转移他们的知识。在大型语言模型成为许多研究人员的关注点后,外源性自我纠正的想法开始引起人们的注意。常见的方法包括使用其他 LLM 作为验证器。验证器通常是较强的 LLM(例如 GPT4),用于验证较小较弱的 LLM 发布的答案的正确性。研究界的共识是,使用外部资源将有助于 LLM 在外部自我纠正期间产生更好的反应[672]。

2) 内源性自我纠正

尽管外源性自我纠正取得了成功,但研究人员对内源性 SC 更感兴趣。原因是外源性

SC 通常需要访问外部知识库，甚至外部更大的 LLM。尽管有研究表明由其他 LLM、编程工具和其他程序组成的多方面框架可以成功地帮助 LLM 生成更好的响应，但这些工具的可用性降低了解决方案的实用性。内源性 SC 的早期可以归因于自我训练的工作，事实证明，如果使用得当，这种方法也将导致更准确的 DL 模型重新思考。内源性 SC 并不是海市蜃楼。If-or-Else(IoE)提示框架[673]，旨在指导 LLM 评估自己的“信心”，促进内在的自我纠正。让 LLM 根据其置信度更新其答案，这项工作取得了出色的结果，其中内源性 SC 有助于提高 4 个不同模型的准确性。

3）步骤 CoT

步骤 CoT 引入了一种新的提示格式 Step CoT Check[674]，它将推理分解为细粒度的步骤。每个步骤都会评估正确性，如果发现错误，系统会提示模型更正错误并结束检查。这种方法通过对推理过程中的每个步骤进行更深入的分析，提高模型检测和纠正错误的能力。这种方法可以产生更好的性能，尤其是在较大的模型中，并且在定位不正确的推理步骤方面特别有效。

6. 压缩输入

在 LLM 的实际应用中，提示至关重要。然而提示技术不可避免地会导致更长的提示，模型的有效上下文长度是有限的，这带来了挑战。为了解决这一挑战，研究者们提出了使用修剪、摘要等压缩方法和检索增强生成方法。

1）思维骨架

人类并不总是按顺序思考问题和编写答案。相比之下，对于许多问题类型，人们首先根据一些协议和策略推导出框架，然后添加证据和细节来解释每一点。在提供咨询、参加考试、写论文等场合尤其如此。思维骨架[675](SoT)在此基础上被提出。具体来说，首先引导 LLM 自己推导出一个骨架。基于骨架，LLM 可以并行完成每个点。SoT 提高了推理效率和否定了完全顺序编码的必要性，并且激发了 LLM 的能力。

2）思维递归

思维递归[676](RoT)，使用了分治的方法。RoT 允许 LLM 通过生成特殊标记来递归地创建多个上下文。即使问题的解决方案超出了最大上下文大小，模型也可以将其划分为多个短上下文，以产生具有极长(100K＋标记)推理步骤的问题。无须任何特定于任务的组件(例如，计算器)，具有 RoT 的模型可以轻松学习解决极其复杂的问题，即使这些问题的解决方案由数十万个 token 组成。

7. 自动化提示

Auto-CoT[677]包括两个主要步骤。首先，将给定数据集的问题划分为几个集群。其次，从每个集群中选择一个代表性问题，并使用 Zero-Shot-CoT 和简单的启发式方法生成其推理链，在推理时使用余弦相似采样找 k 个相似问题再进行回答。

Auto-Reason[654]框架由几个关键组件组成，初始查询被假定为零样本提示，首先使用包含多个思维链示例的提示模板进行格式化。这个精心设计的提示旨在通过采用 CoT 策略从 LLM 中引出原理，鼓励模型将问题分解为一系列明确的推理步骤，然后输入 LLM 中。接着，根据提供的提示生成原理。随后使用另一个提示模板对这些原理进行格式化以获得最终答案。Auto-Reason 框架的一个关键优势是它能够利用提供的提示模板适应各种 LLM。

5.1.4 基于数据集构建的方法

创建大规模、高质量的推理数据集对于增强 LLM 的推理能力至关重要。人工注释被广泛认为是高质量的,但涉及成本问题。使用 LLM 自动化注释过程提供了一种更具成本效益的替代方案。

1. 人工构建方法

人工注释在为 LLM 构建数据集中的作用是必不可少的,人工注释更为细致和准确,并且更能处理模糊的问题,Zhou 等的研究[678]揭示了人工注释的数据在增强大型语言模型的推理能力方面起着关键作用。

增强 LLM 中的推理能力需要过程监督,其中人工注释者指导推理过程的每个步骤。然而,这种监督需要大量的人工标注。仅通过人工标注构建数据集变得越来越不切实际,这凸显了需要替代方法来改进推理。一种很有前途的方法是人类和 LLM 协作标注,即使用 LLM 来执行第一轮标注,在细化阶段,使用人工标注评估 LLM 生成的注释的质量,并专注于仅更正质量较差的注释子集。与仅依赖人工的传统标注相比,LLM 通常可以更快、更低成本生成 labels,人工标注者可以评估 LLM 生成的注释的质量,并专注于仅更正质量较差的注释[679]。最近的工作越来越关注如何在确保数据质量的同时最大限度地提高自动化程度,从而在不影响注释准确性的情况下减少人工参与[680]。

2. 自动化构建

人工构建数据集任务通常很乏味、耗时,并且需要大量的人力和资金。一方面的研究[662]认为 LLM 是零样本推理机,设计零样本思维链,使用与任务无关的单一的提示(如 Let's think step by step)即可在数学方面达到较好的性能。一方面数据标注是一项具有挑战性且耗费大量资源的任务,尤其是在需要筛选、识别、组织和重建文本数据等复杂操作的场景中。人类为设计特定任务所做的努力甚至更多:不同的任务需要不同的方式。利用 LLM 进行数据注释提供了一种经济且高效的替代方案。最初的自动化方法聘请外部更强大的 LLM 来注释由较小的 LLM 生成的中间过程。此外,基于蒙特卡洛的方法减少了对外部更强 LLM 的依赖,可以使用较弱的 LLM 来完成数据注释,从而通过自我强化的方式训练更强的 LLM。

3. 基于 LLM 的数据标注方法

使用更强的 LLM 进行数据标注,指的是利用这些大型预训练的模型自动为数据集中的文本添加注释、标签或分类信息。这样的做法可以帮助快速处理大规模的数据集,尤其是在情感分析、实体识别、文本分类等任务中。相比人工标注,LLM 可以大大加速注释过程,特别是在面对大规模数据时。LLM 能保持注释的一致性,并且节省成本。通过利用更强大的外部模型的功能,这种方法提高了标记过程的准确性和可扩展性,使其更适合大规模任务。然而这种方法过于依赖外部 LLM 的性能,可能继承其思维偏差,数据隐私和安全问题的风险也需要考虑在内。

5.1.5 基于蒙特卡洛的方法

为了实现最佳质量,需要训练数据来构建强大的验证器模型。手动收集解决方案 labels 成本高昂且不可扩展。Wang 等[681]建议在中间解的完成时使用蒙特卡洛抽样来获得

逐步训练标注。具体来说，对于每个中间解决方案，通过样本解码机制使用推理器多次完成解决方案，完成的解决方案正确的百分比称为解决方案的正确性，这种暴力方法需要大量的策略调用。

使用二分搜索的蒙特卡洛[682]方法对单纯蒙特卡洛的方法进行了优化，将解答路径表示为状态-动作树，每个状态 s 表示问题和解答的部分步骤，每个动作 a 表示可能的动作。通过语言模型生成步骤，形成完整的推理路径。采用蒙特卡洛估计 MC(s)评估状态的正确性，并利用状态-推理价值函数 $Q(s,r)$评估 rollout 价值。再采用改进的 PUCT 公式进行节点选择，优先选择较少访问但估值较高的 rollouts，从而减少计算开销并提高效率。这种方法在数学问题解决方面表现出卓越的性能。

Zhang 等[683]在 Process Annotation 研究中，进一步推动了基于 MCTS 的模拟，提出了一种自优化机制。该方法利用获取的流程注释来训练流程奖励函数(PRM)，以提升 LLM 的性能。随后，优化后的 LLM 被用于重复 MCTS 模拟，从而生成更高质量的注释。该迭代过程通过不断地优化循环，使流程注释逐步改进和完善。实验结果表明，该方法在数学问题求解、问答、多领域知识推理等任务中表现出色，验证了其在不断迭代增强下对标注质量的持续优化能力。

5.1.6　逆转诅咒的缓解方法

Guo 等[684]的研究表明，使用 COT 方法几乎无法解决逆转诅咒，更进一步地，轻量级的方法都几乎无法解决逆转诅咒。由于 Zipf 定律[685]，在互联网上，许多事实很少被提及，或者只被提及一次(因此是单向的)，这更促进了研究者们在不同层面上对 LLM 进行优化训练。

1. 重排训练

LLM 是从左到右以自回归方式训练的，重排训练有可能解决逆转诅咒，因为它允许模型看到反向的事实。事实上，LLM 足够强大，可以理解对称关系，难点在于回忆起前向的词[684]。BERT 及其变体在许多自然语言理解任务上取得了显著的成绩，但它们在某些任务上实际上并不依赖于词序信息。这些模型在一定程度上类似于词袋模型(Bag-of-Words，BOW)，并且在许多任务中，词序的随机打乱对其预测结果几乎没有影响[695]。CoT 几乎无法缓解反转失败[684]。在此理论基础上，许多研究提出了对数据集进行各种各样的重排训练。

1) 逆向训练

小成本逆向训练主要包括 token 反转、单词反转、实体翻转、随机段反转，其中正向和反向训练样本被随机排列。在单词反转中，基本上是从最后一个单词开始向后(从右到左)预测句子。在随机段反转中，段由[REV]分隔，因此训练标记的数量增加了一倍。逆向转换可以看作是模型必须学习的第二种“语言”。实验证明，以上反转方法都不会影响模型在正向任务的性能，除此外，相比先使用知识进行预训练再使用逆向数据进行微调，指令调整数据添加到预训练级别通常表现更好[686]，这种构造数据集的方法较为简单而且没有使用额外的数据结构。

2) 语义感知排列训练

对反转诅咒进行了全面的评估和分析，研究[684]认为根本原因主要在于训练阶段和推

理阶段之间的词序不同。语义感知排列训练(SPT)方法是通过将句子分割成语义单元(如短语或实体)来增强训练过程的方法。SPT 应用 3 个不同的顺序来排列这些块:原始顺序、反转顺序和随机排列顺序。采用辅助模型将训练句子分割成几个最小的语义单元,然后重新排序以馈送到模型中。实验表明,由 SPT 训练后,模型在反向问题上的表现几乎与在前向问题上一样好,从而有效地缓解了反转诅咒。

2. 演绎闭合训练

演绎闭合训练(Deductive Closure Training,DCT),旨在通过推理提高语言模型的准确性、连贯性和可更新性。与传统的训练方法不同,对于给定一组种子文档,DCT 通过语言模型生成与这些文档相关或矛盾的其他文本。然后,模型会对这些生成的文本进行推理,判断哪些部分最可能是真实的,并基于这一推理结果进行微调,从而提高模型的连贯性和准确性。在监督学习与无监督学习上,DCT 均取得了更高的准确性[687]。

DCT 可以以多种不同的方式应用,具体取决于种子文档的来源。如果这些来自可信的事实来源,则 DCT 可用于对事实进行监督改编。如果文档包含要插入 LM 中的新信息,DCT 会提供用于模型更新的工具,如果种子文档是由模型本身生成的,则 DCT 可以对模型进行完全无监督的微调,以提高准确性。

3. 散列掩码

散列方法是一种新的掩码方法,散列方法不同于传统的 LLM 掩码方法,它使用无意义标识符,用唯一的散列值替换文本,允许在整个文本中被引用,散列方法优化了自由文本和表格格式的表现,并提高模型在涉及逻辑推理和统计学习的各种任务中的性能[688]。

4. 语言模型编辑

Transformer 模型(如 BERT、GPT 等)在自然语言处理任务中取得了显著的成就,其中一个关键因素是它们能够通过大量的参数隐式记忆事实知识。然而,如何让这些模型“忘记”某些已存储的知识并“记住”新的知识,尤其是在保持模型性能不下降的情况下,这个问题尚未得到充分研究[689]。许多编辑方法和 LLM 虽然在编辑方向上有效地回忆编辑事实,但在反向评估时存在严重缺陷。目前的编辑方法远未完全理解注入的知识,而只是通过硬编码简单地记忆。双向可逆关系建模(BIRD)[689]方法通过修改模型的权重,捕捉事实知识中主语与宾语之间的正向和反向关系(即以“主语,关系,宾语”三元组表示的事实知识)。BIRD 在编辑过程中将双向可逆的关系融入模型,即一对一、一对多、多对一和多对多的基本关系,帮助模型在两种方向上更好地泛化和回忆事实。

5. 文本上使用非自回归模型

自回归模型(ARM)多年来一直主导着语言建模领域。但是自回归模型的顺序采样过程效率低下,限制了非顺序的推理能力。从本质上讲,这是因为此类模型通过概率链式法则来表征联合分布,从而激发了开发其他类型的文本生成模型的研究[690],其中主要的是使用扩散模型代替自回归模型。主要技术是对熵离散扩散(SEDD),SEDD 采用离散状态(吸收)马尔可夫过程,该过程通过用掩码令牌随机替换令牌来向数据添加噪声然后学习一个相反的过程,从一个完全被掩盖的句子中去噪。SEDD 在 GPT-2 规模的 5 个零镜头语言建模基准上取得了与自回归模型相当的结果。同时,SEDD 可以减少采样中的函数评估(NFE)数量,并实现以不同位置的提示为条件的文本。