Chain of Draft 论文解读

2025年3月1日 3 分钟阅读

#AI #Chain of Draft #论文 #技术

> *本文介绍了Chain of Draft（CoD）论文，并对其技术原理、主要贡献、论文方法、评估结果和局限性进行了详细解读。*

https://arxiv.org/abs/2502.18600

本研究提出了一种名为“草稿链”（Chain of Draft, CoD）的新型提示策略，旨在提高大型语言模型（LLM）在复杂推理任务中的效率。CoD 模仿人类的认知过程，鼓励 LLM 生成简洁但信息丰富的中间推理结果，而不是像“思维链”（Chain-of-Thought, CoT）那样产生冗长的步骤。实验结果表明，CoD 在保持或提高准确性的同时，显著减少了 token 使用量和延迟，从而降低了计算成本。 这种方法特别适用于对成本和延迟敏感的实际应用场景。CoD 通过减少不必要的文字，专注于关键见解，优化了 LLM 的推理过程。 研究还探讨了 CoD 对 LLM 设计、部署和实际可用性的影响，并展望了未来结合 CoD 与其他优化技术的可能性。

论文由Zoom Communications机构提供。论文的作者包括Silei Xu、Wenhao Xie、Lingxiao Zhao和Pengcheng He。

论文十问解读

论文试图解决什么问题？ 这篇论文旨在解决大型语言模型（LLMs）在复杂推理任务中使用**Chain-of-Thought (CoT)** 方法时产生的**冗长和高计算成本**问题。CoT 虽然提高了准确性，但其详细的逐步推理过程导致大量的 token 使用和更高的延迟，这在实际应用中是不利的。因此，论文提出了 **Chain of Draft (CoD)**，一种更高效、简洁的推理提示策略。

研究现状如何？ 目前，LLMs 通过 **CoT** 等结构化推理方法在复杂任务中表现出色。一些研究通过 **self-consistency CoT, ReAct** 等方法来增强推理的可靠性。然而，这些方法增加了 token 的使用量，使其难以应用于对成本和延迟敏感的场景。诸如 **Skeleton-of-Thought (SoT)** 和其他降低延迟的技术，要么不能减少计算成本，要么在复杂任务中效果不佳。**Concise**

目录

文章信息

字数

阅读时间

发布时间

更新时间

标签

#AI #Chain of Draft #论文

Thoughts (CCoT) 和 **token-budget-aware LLM reasoning (TALE)** 等方法试图通过限制 token 数量来提高效率，但它们在token预算的动态调整和复杂任务的适应性方面存在局限性。

论文提出了什么方法?

论文提出了 **Chain of Draft (CoD)** 提示策略，灵感来源于人类在解决复杂问题时采用的简洁记录关键信息的习惯。

CoD 鼓励 LLMs 在每个推理步骤中生成简洁、信息密集的输出，从而减少冗余，降低延迟和计算成本。

CoD 的核心思想是限制每个推理步骤的字数，使其只关注必要的计算或转换。例如，在解决数学问题时，CoD 会将推理过程简化为简洁的方程式。

论文如何验证所提出的方法?

论文在多个需要多步骤推理的 benchmark 上进行了实验，包括算术推理 (**GSM8k**)、常识推理 (来自 **BIG-bench** 的日期理解和体育理解) 和符号推理 (coin flip tasks)。

算术推理：使用了 GSM8k 数据集。该数据集包含 8500 个小学水平的数学问题，涵盖算术、几何、代数和逻辑推理。

常识推理：使用了 BIG-bench 中的日期理解和体育理解任务。

符号推理：使用了论文中介绍的 coin flip tasks（抛硬币任务）。由于确切的数据集未公开发布，研究人员合成了包含 250 个示例的测试集。

实验使用了 **GPT-4o** 和 **Claude 3.5 Sonnet** 这两个流行的模型。

对比了 **CoD**、**CoT** 和标准提示 (Standard prompting) 三种策略。

评估指标包括准确率、token 使用量和延迟。

实验结果如何?

在所有测试中，**CoD** 在保持或提高准确性的同时，显著减少了 token 的使用量和延迟。

例如，在 **GSM8k** 算术推理任务中，**CoD** 在 token 使用量减少 80% 的情况下，准确率与 **CoT** 相当。

在体育理解任务中，**CoD** 将 **Claude 3.5 Sonnet** 的平均输出 token 从 189.4 减少到 14.3，减少了 92.4%。

在 coin flip 任务中，**CoD** 和 **CoT** 都达到了 100% 的准确率，但 **CoD** 显著减少了 token 的使用。

论文有哪些重要的结论?

CoD 是一种有效的推理方法，可以在不牺牲准确性的前提下显著降低延迟和计算成本。

简洁的推理步骤可以提高 LLMs 的效率，使其更适用于实际应用。

CoD 表明，有效的推理不一定需要冗长的输出，可以在保持推理深度的同时最小化冗余。

论文有哪些局限性？ 论文中提到，**CoD** 策略在提示中指导模型限制每个推理步骤的字数，但并没有强制执行这种限制。这可能导致在某些情况下，模型仍然会生成较长的推理步骤。

未来的研究方向有哪些？

将 **CoD** 与其他降低延迟的方法 (如自适应并行推理或多通道验证) 相结合，进一步优化性能。

通过使用简洁的推理数据进行训练，改进推理模型，同时保持 LLMs 的可解释性和效率。

这篇论文对领域有什么贡献？ 这篇论文的主要贡献在于提出了 **Chain of Draft (CoD)** 这一新的提示策略，它通过模仿人类简洁的思考方式，显著降低了 LLMs 在推理过程中的计算成本和延迟，同时保持了较高的准确性。这为 LLMs 的实际应用提供了一种更高效的解决方案。

是否有其他相关的研究？ 论文中提到了很多相关研究，包括 **self-consistency CoT**、**ReAct**、**Skeleton-of-Thought (SoT)**、**Concise Thoughts (CCoT)** 和 **token-budget-aware LLM reasoning (TALE)**。这些研究都致力于提高 LLMs 的推理能力和效率，但各有侧重。

论文CoD的实例

这篇论文中提供了一个**Chain-of-Draft (CoD)**的实际例子，展示了它与标准提示 (Standard prompting) 和**Chain-of-Thought (CoT)**提示策略的不同。

以下是这个例子：

问题 (Q)：Jason 有 20 根棒棒糖。他给了 Denny 一些棒棒糖。现在 Jason 有 12 根棒棒糖。Jason 给了 Denny 多少根棒棒糖？

标准提示 (Standard)：直接输出答案，没有任何推理过程。

回答 (A): 8

Chain-of-Thought (CoT)：提供详细的推理过程，但包含不必要的细节。

回答 (A)：让我们逐步思考：1. 最初，Jason 有 20 根棒棒糖。2. 给 Denny 一些后，Jason 现在有 12 根棒棒糖。3. 为了找出 Jason 给了 Denny 多少根棒棒糖，我们需要计算初始棒棒糖数量和剩余数量之间的差额。4. 我们可以建立一个简单的减法问题：初始棒棒糖数量 - 剩余棒棒糖数量 = 给 Denny 的棒棒糖数量 5. 放入数字：20 - 12 = 给 Denny 的棒棒糖数量 6. 解减法：20 - 12 = 8 因此，Jason 给了 Denny 8 根棒棒糖。#### 8 根棒棒糖

Chain-of-Draft (CoD)：将推理过程浓缩为最小的、抽象的表示，只关注必要的数学运算。

回答 (A)：20 - x = 12; x = 20 - 12 = 8. #### 8

在这个例子中，**CoD** 通过使用简洁的方程式，**直接表达了解题所需的关键信息，避免了冗长的解释和不必要的细节。**这使得 **CoD** 既保持了透明度和正确性，又显著减少了 token 数量。

CoD 与 CoT 的对比

Prompt 示例：

```
# Standard
Answer the question directly. Do not return any preamble, explanation, or reasoning.

# Chain-of-Thought
Think step by step to answer the following question. Return the answer at the end of the response after a separator ####.

# Chain-of-Draft
Think step by step, but only keep a minimum draft for each thinking step, with 5 words at most. Return the answer at the end of the response after a separator ####.
```

CoD 和 CoT 在不同模型上的效果对比表格：

任务	模型	Prompt	准确率	Token 使用量	延迟(秒)
算术推理	GPT-4o	CoT	95.4%	205.1	4.2
		CoD	91.1%	43.9	1.0
	Claude 3.5 Sonnet	CoT	95.8%	190.0	3.1
		CoD	91.4%	39.8	1.6
常识推理(日期理解)	GPT-4o	CoT	90.2%	75.7	1.7
		CoD	88.1%	30.2	1.3
	Claude 3.5 Sonnet	CoT	87.0%	172.5	3.2
		CoD	89.7%	31.3	1.4
常识推理(体育理解)	GPT-4o	CoT	95.9%	28.7	0.9
		CoD	98.3%	15.0	0.7
	Claude 3.5 Sonnet	CoT	93.2%	189.4	3.6
		CoD	97.3%	14.3	1.0
符号推理(抛硬币)	GPT-4o	CoT	100.0%	52.4	1.4
		CoD	100.0%	16.8	0.8
	Claude 3.5 Sonnet	CoT	100.0%	135.3	3.1
		CoD	100.0%	18.9	1.6

从以上数据可以看出，**CoD 在保持或提高准确性的同时，显著减少了 token 的使用量和延迟。**这表明 CoD 是一种更高效的推理策略。

相对于 Chain of Draft (CoD)，Chain-of-Thought (CoT) 存在以下缺陷：

> **冗长性**：CoT 提供详细的推理过程，但通常包含不必要的细节。例如，在解决数学问题时，CoT 可能会包含与解题无关的背景信息。这种冗长性导致 token 数量增加，计算成本上升.
>
> **高延迟**：由于 CoT 需要生成较长的推理步骤，因此响应时间较长。这在对延迟敏感的实际应用中是一个显著的缺点.
>
> **资源消耗**：CoT 方法需要更多的计算资源，因为它需要处理大量的 token. 即使是简单的任务，CoT 也可能导致过度思考，从而浪费资源。

缺乏效率：CoT 在生成推理步骤时，没有优先考虑效率和简洁性。这与人类解决问题的方式不同，人类通常会使用简洁的草稿或笔记来捕捉关键信息.

成本较高：由于 CoT 需要生成大量的 token，因此在使用 LLM 时会产生更高的成本。这在需要大规模部署 LLM 或预算有限的情况下是一个重要考虑因素.

总而言之，CoT 的主要缺陷在于其**冗长性、高延迟、资源消耗和成本较高。**CoD 通过鼓励 LLM 生成简洁、信息密集的推理步骤，**在保持或提高准确性的同时，显著减少了 token 的使用量和延迟。**

论文中提到的其他方法

论文中提到了几个CoT类似的理论，并在论文中分析了它们的优缺点。这些理论包括：

Concise Thoughts (CCoT)：CCoT 建议为推理步骤使用固定的全局 token 预算。

优点：旨在通过限制 token 数量来提高效率。

缺点：不同的任务可能需要不同的预算才能在性能和成本之间取得最佳平衡。LLM 可能无法遵守不切实际的预算，经常生成比预期更多的 token。

Token-budget-aware LLM reasoning (TALE)：TALE 通过动态估计不同问题的全局 token 预算来扩展 CCoT 的思想，预算基于推理的复杂性。

优点：试图根据任务的复杂性动态调整 token 预算，从而更有效地利用资源。

缺点：需要额外的 LLM 调用来估计预算，这会增加延迟。它假设模型可以准确预测请求的复杂性，这限制了其在更复杂任务中的适用性，在这些任务中，可能需要在推理过程中进行反思、自我纠正或外部知识检索。

Skeleton-of-Thought (SoT)：SoT 首先引导 LLM 生成答案的骨架大纲，然后进行并行解码以减少延迟。

优点：有助于降低延迟，通过并行解码加速生成过程。

缺点：不降低计算成本，仅限于可以有效并行化的任务。

Continuous Latent Space Reasoning (Coconut)：Coconut 训练 LLM 在连续潜在空间中执行推理，而不是在传统的自然语言空间中使用 LLM 的最终隐藏状态来表示推理过程.

优点：减少延迟和计算成本。

> **缺点**：在复杂任务（如 GSM8k）中，准确性降低。此外，它失去了自然语言推理的可解释性，并且不能应用于像 GPT 和 Claude 这样的黑盒模型。

总的来说，这些方法都试图在推理的准确性和效率之间找到平衡，但各有优缺点。**CoD 的优势在于它采用了一种更灵活的策略，允许每个步骤有不同的 token 预算，从而更好地适应各种结构化推理技术。**

论文不足之处

不足之处主要体现在以下几个方面：

CoD 策略的约束力不足：论文提到，**CoD 策略在提示中指导模型限制每个推理步骤的字数，但并没有强制执行这种限制。**这可能导致在某些情况下，模型仍然会生成较长的推理步骤，从而影响 **CoD** 的效率。虽然设定了每个推理步骤最多五个单词的指导方针，但实际上并没有强制执行这一限制。

实验数据集的局限性：虽然论文在算术推理 (**GSM8k**)、常识推理 (来自 **BIG-bench** 的日期理解和体育理解) 和符号推理 (coin flip tasks) 等多个任务上进行了评估，但这些数据集可能无法完全代表所有类型的复杂推理任务。例如，**coin flip tasks** 使用的是研究人员自己合成的数据集，可能存在一定的偏差。

缺乏与其他延迟降低方法的结合：论文提出，未来的工作可以将 **CoD** 与其他降低延迟的方法 (如自适应并行推理或多通道验证) 相结合，以进一步优化性能。然而，论文本身并没有进行这方面的实验，因此无法证明 **CoD** 与其他方法结合的实际效果。

没有提供相应的 GitHub 项目：论文主要介绍了 **CoD** 的概念、实验结果和潜在应用，但没有提及任何相关的代码库或项目发布。这使得其他研究人员难以复现论文的结果或将 **CoD** 应用于自己的项目中。

分享这篇文章

相关文章推荐

Test-Time Scaling 相..

本文介绍了Test-Time Scaling（...

DeepSeek V3 论文解读

本文介绍了深度求 ...

DeepSeek 微调

本文介绍了如何使用合成推理...