Google DeepMind

Chain-of-Thought Reasoning without Prompting

Xuezhi Wang[1] **and Denny Zhou**[1]
[1]Google DeepMind, [1]{xuezhiw, dennyzhou}@google.com

分享人：丁伯瑞

2024.12.24

NIPS2024

- 作者介绍
- 研究背景
- 本文方法
- 实验结果

作者介绍

Xuezhi Wang

关注

Research Scientist, Google DeepMind
在 google.com 的电子邮件经过验证 - 首页

Machine Learning Natural Language Processing

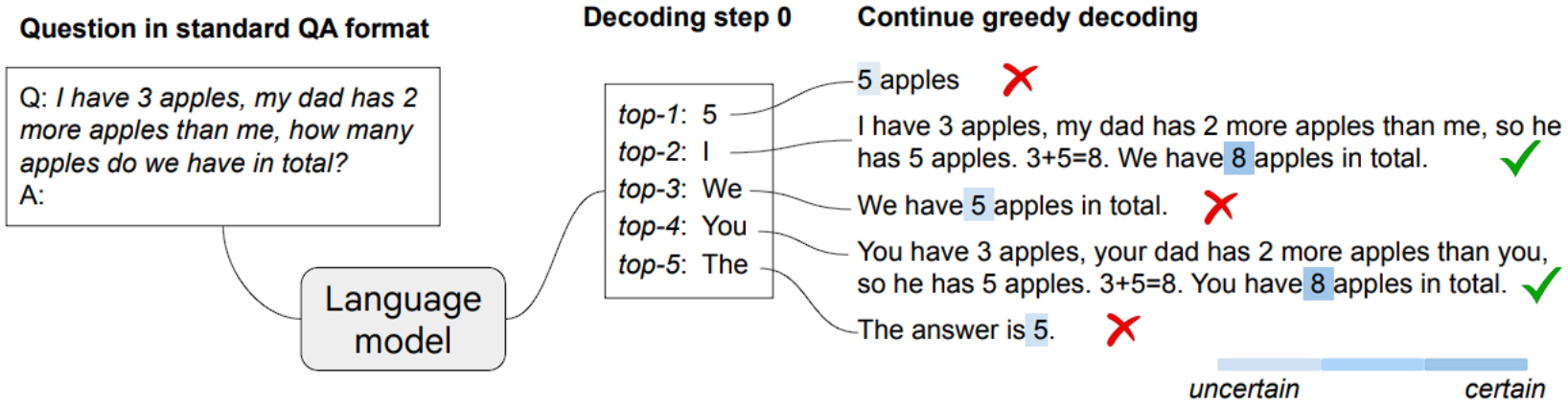
标题	引用次数	年份
Chain of thought prompting elicits reasoning in large language models J Wei, X Wang, D Schuurmans, M Bosma, E Chi, Q Le, D Zhou Neural Information Processing Systems (NeurIPS), 2022	10499 *	2022
Palm: Scaling language modeling with pathways A Chowdhery, S Narang, J Devlin, M Bosma, G Mishra, A Roberts, ... Journal of Machine Learning Research (JMLR), 2023	5372	2023
Scaling instruction-finetuned language models HW Chung, L Hou, S Longpre, B Zoph, Y Tay, W Fedus, Y Li, X Wang, ... JMLR 2024	3214	2024
Self-consistency improves chain of thought reasoning in language models X Wang, J Wei, D Schuurmans, Q Le, E Chi, S Narang, A Chowdhery, ... ICLR 2023	2552 *	2023

- 作者介绍
- 研究背景
- 本文方法
- 实验结果

研究背景

- 引发LLM推理能力的两种方法：
  - 由一些prompt引发的：一些带有中间推理步骤的少样本prompt，或者是要求输出中间推理步骤的零样本prompt。
  - 使用大量的思想链推理数据进行模型训练或指令微调。

- LLM能否在没有提示的情况下有效地进行推理？



- 作者介绍
- 研究背景
- 本文方法
- 实验结果

方法介绍

- 置信度

$$\Delta_{k,\text{answer}} = \frac{1}{|\text{answer}|} \sum_{x_t \in \text{answer}} p(x_t^1 \mid x_{<t}) - p(x_t^2 \mid x_{<t}).$$

$\mathbf{x}_t^1$和$\mathbf{x}_t^2$分别表示第K条解码路径中第t步解码的前两个词元，这两个词元是基于其在softmax后的最大概率从词汇表中选择的，前提是x_t属于答案词元。

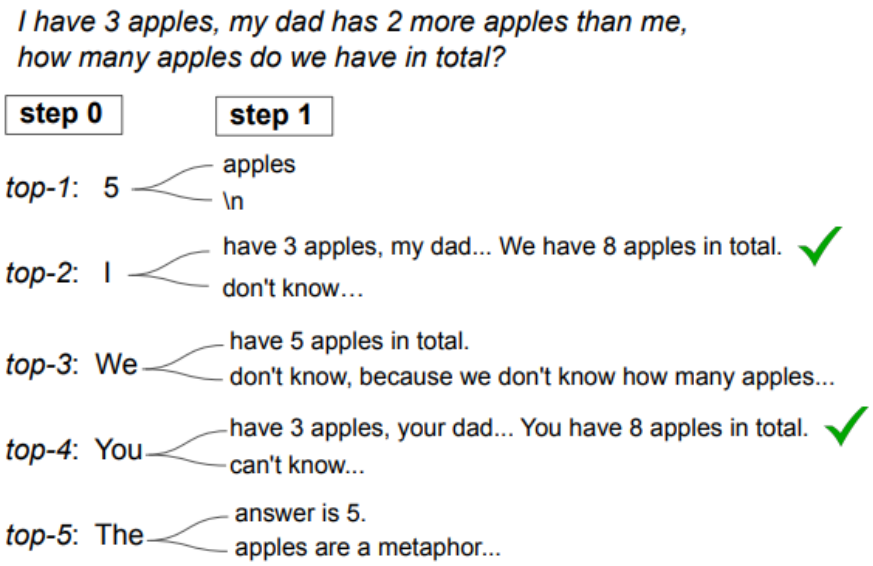
方法介绍

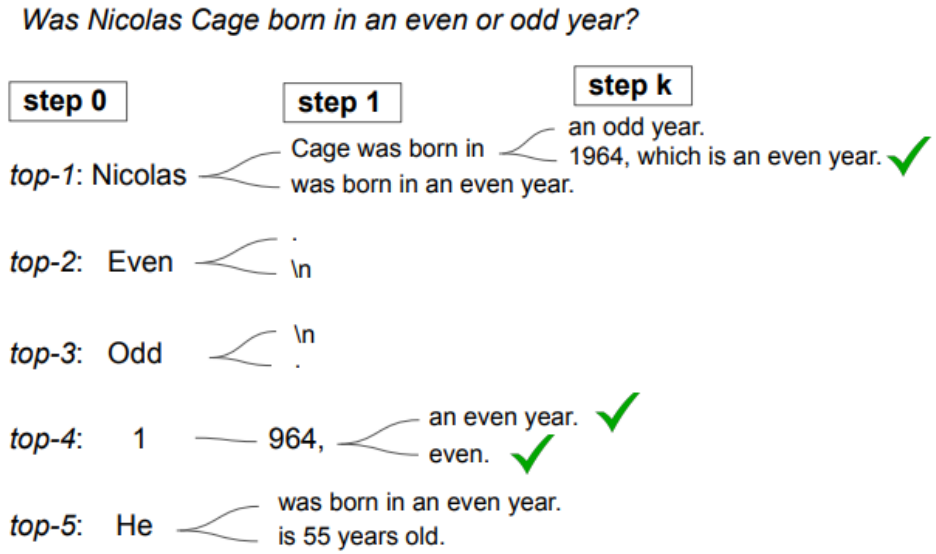
[GSM8K] *Kylar went to the store to buy glasses for his new apartment. One glass costs $5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?*
Greedy path:
$k = 0$: \$**60.00** (0.029)
Alternative top-k paths:
$k = 1$: **60** (0.058)
$k = 2$: Kylar needs to pay \$**60** for 16 glasses. (0.058)
. . .
$k = 7$: If Kylar buys 16 glasses, he will pay \$**60**. (0.032)
$k = 9$: We can calculate the price of 16 glasses by multiplying the price of one glass by 16. However, we need to remember that every second glass costs only 60% of the price. Therefore, we need to multiply the price of one glass by 16 and then subtract 40% of the price of 8 glasses. 16 x 5 = 80 8 x 5 = 40 40 x 0.4 = 16 80 – 16 = 64 Kylar needs to pay \$**64** for 16 glasses. (0.994)

[Year Parity] *Was Nicolas Cage born in an even or odd year?*
Greedy path:
$k = 0$: Nicolas Cage was born in an **odd** year. (0.117)
Alternative top-k paths:
$k = 1$: **Even** (0.207)
$k = 2$: **Odd** (0.198)
$k = 3$: 1964, an **even** year. (0.949)
$k = 4$: He was born in an **even** year. (0.0)
. . .
$k = 7$: Cage was born in 1964, an **even** year. (0.978)

Table 1 | Examples of greedy decoded paths and alternative top-k paths over the PaLM-2 Large model. The model's confidence over the answers (bolded) are highlighted in blue (See §2.2 for details).

方法介绍





propose a weighted aggregation method, i.e., we take the answer that maximizes $\tilde{\Delta}_a = \sum_k \Delta_{k,a}$ where $\Delta_{k,a}$ is the k-th decoding path whose answer = a. We found that adopting this approach enhances the stability of the results, and further analysis is presented in Section §3.3.

- 作者介绍
- 背景介绍
- 方法介绍
- 实验结果

实验结果

	GSM8K (top-100)	Year Parity
Greedy decoding	44.0%	57.0%
Decode 10 paths, rank by model's highest log-prob	37.0%	55.0%
Decode 10 paths, rank by model's highest length-normalized log-prob	51.0%	57.0%
CoT-decoding (decode 10 paths, rank by model's answer confidence)	**72.0%**	**95.0%**

Table 2 | CoT-decoding reliably extracts the CoT-paths compared to other methods (on PaLM-2 L).

- Log-prob(对数概率)：

$$\log P(Y) = \sum_{i=1}^{n} \log P(y_i | y_1, y_2, \ldots, y_{i-1})$$

- Length-normalized log-prob(长度归一化概率)：

$$\text{Normalized log-prob} = \frac{\log P(Y)}{n}$$

  - 防止较长的路径因为概率值累乘而显得过低，归一化处理可以使长短路径的对数概率在比较时更为公平。

实验结果

Table 3 | CoT-decoding and self-consistency w/o prompts on GSM8K.

	Mistral-7B	PaLM-2 L
Greedy decoding	9.9%	34.8%
Self-consistency without CoT-prompt (10 paths)	12.9%	40.6%
CoT-decoding (10 paths)	**25.1%**	**63.2%**

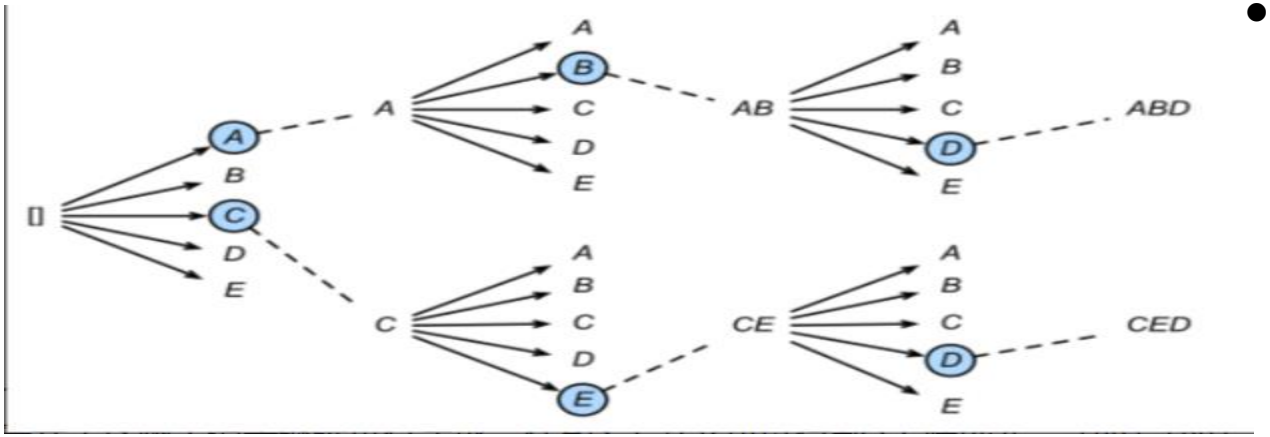
- 贪婪解码：每次选取概率最大的词输出
- 自我一致性：给出多个cot的prompt输出多个解释，选取答案相同的最多票数的答案

```
Prompt: 问：小树林里有15棵树。树林工人今天将在树林里种树。他们完成后将有21棵树。林场工人今天种了多少棵树？
        答：我们开始有15棵树。后来我们有21棵树。差额一定是他们种的树的数量。所以，他们一定是种了21-15=6棵树。答案是6。
        问：如果停车场里有3辆汽车，又有2辆汽车到达，那么停车场里有多少辆汽车？
        答：停车场里已经有3辆汽车。又有2辆到达。现在有3+2=5辆车。答案是5。
        问：利亚有32块巧克力，她姐姐有42块。如果她们吃了35块，她们总共还剩下多少块？
        答：利亚有32块巧克力，利亚的姐姐有42块。这意味着原来有32+42=74块巧克力。35块已经被吃掉了。所以他们总共还有74-35=39块巧克力。答
        问：杰森有20根棒棒糖。他给了丹尼一些棒棒糖。现在杰森有12根棒棒糖。杰森给了丹尼多少个棒棒糖？
        答：杰森有20根棒棒糖。因为他现在只有12个，所以他肯定把剩下的给了丹尼。他给丹尼的棒棒糖的数量一定是20-12=8根。答案是8。
        问：肖恩有五个玩具。圣诞节时，他从他的妈妈和爸爸那里各得到两个玩具。他现在有多少个玩具？
        答：他有5个玩具。他从妈妈那里得到了2个，所以之后他有5+2=7个玩具。然后他又从爸爸那里得到了2个，所以他总共有7+2=9个玩具。答案是9.
        问：机房里有九台电脑。从星期一到星期四，每天都有五台电脑被安装。现在机房里有多少台电脑？
        答：从星期一到星期四有4天。每天增加5台电脑。这意味着总共4*5=20台电脑。一开始有9台电脑，所以现在有9+20=29台电脑。答案是29。
        问：迈克尔有58个高尔夫球。在星期二，他丢了23个高尔夫球。星期三，他又丢了两个。他在周三结束时有多少个高尔夫球？
        答：迈克尔最初有58个球。周二他丢了23个，所以之后他有58-23=35个球。在周三他又丢了2个，所以现在他有35-2=33个球。答案是33。
        问：奥利维亚有23美元，她买了五个面包圈，每个3美元。她还剩下多少钱？
        答：她买了5个面包圈，每个3美元。这意味着她花了5*3=15美元。由于她原来有23美元，现在花了15美元，那么还剩23-15=8美元。答案是8。
        问：当我6岁时，我妹妹的年龄是我的一半。现在我70岁了，我的妹妹现在多大？
        答：
Output1: 当你6岁时，你妹妹的年龄是你的一半，也就是3岁。现在你是70岁，相当于经过了64年。因此你妹妹现在应该是3+64=67岁。答案是67。
Output2: 当你6岁时，你妹妹的年龄是3岁（一半）。现在经过64年（70-6），你妹妹也长大了64岁。所以现在你妹妹应该是3+64=67岁了。答案是67。
Output3: 6岁时，我妹妹的年龄是我的一半，也就是3岁。70岁时，我妹妹的年龄应该是70/2=35岁。答案是35岁。
```

实验结果

Table 4 | CoT-decoding is the only decoding strategy that can significantly enhance language models' reasoning.

	GSM8K Acc
Top-k sampling ($k = 10$)	4.9%
Top-p / Nucleus sampling ($p = 0.9$)	6.4%
Beam search ($b = 10$)	6.7%
Temperature sampling ($T = 0.7$)	7.5%
Greedy decoding	9.9%
Self-consistency w/o CoT prompt (10 paths)	12.9%
CoT-decoding ($k = 10$)	**25.1%**



- Top-k 采样：从 tokens 里选择 k 个作为候选，然后根据它们的 likelihood scores 来采样模型从最可能的“k”个选项中随机选择一个。
- Top-p采样：从 tokens 里选择 n 个概率和到p作为候选，然后根据它们的 likelihood scores 来采样模型从最可能的“n”个选项中随机选择一个。
- 温度采样：通过温度，在采样前调整每个词的概率分布。温度越低，概率分布差距越大，越容易采样到概率大的字。温度越高，概率分布差距越小，增加了低概率字被采样到的机会。
- 束搜索：

实验结果

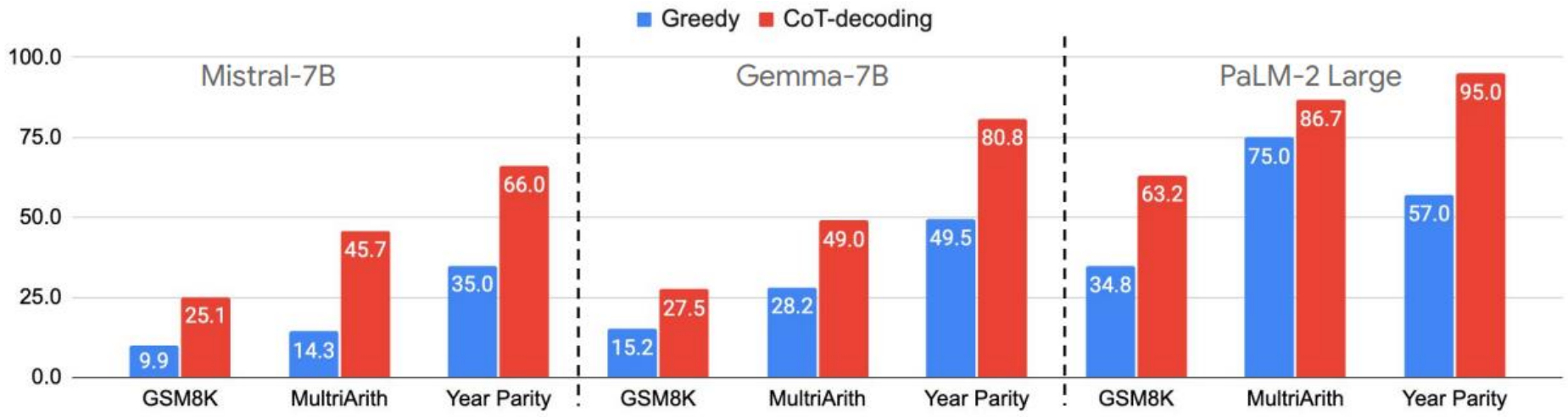
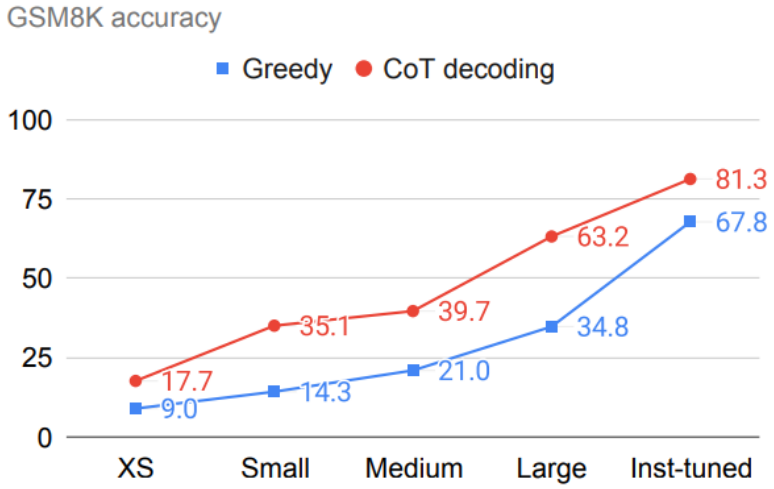


Figure 3 | CoT-decoding effectively elicits reasoning across multiple language model families including PaLM-2, Mistral and Gemma, with significant accuracy gains over three reasoning tasks.

实验结果



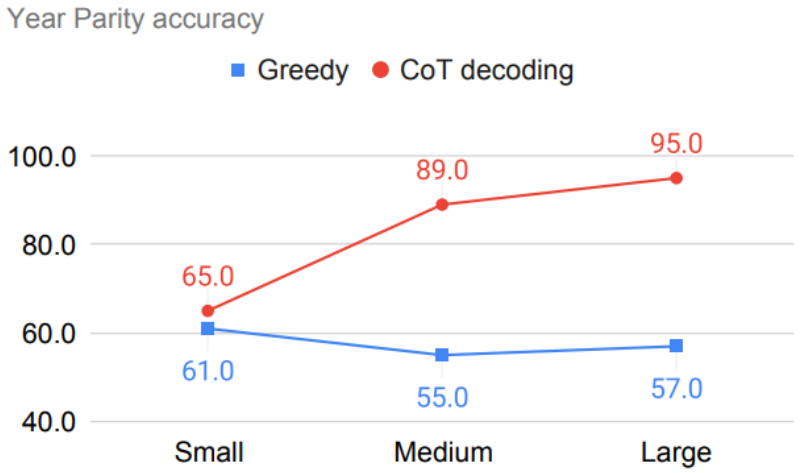
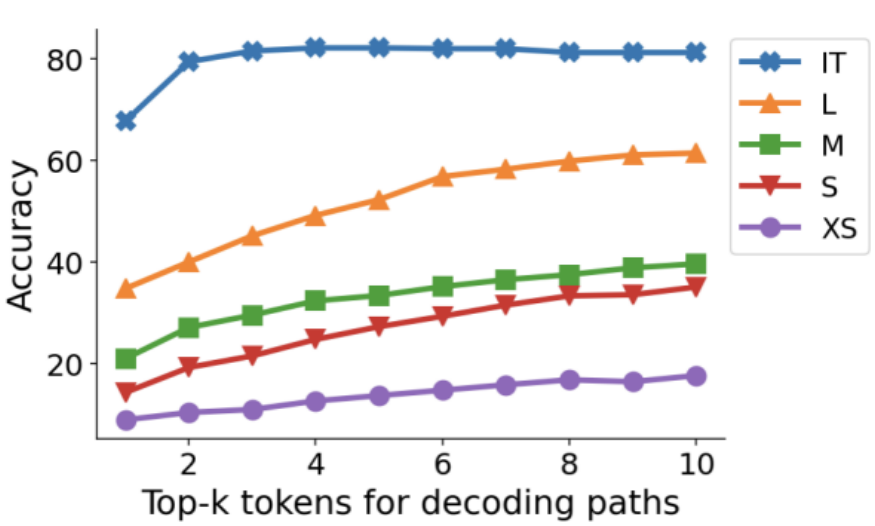


Figure 4 | CoT-decoding reliably improves reasoning performance across model scales (PaLM-2), even when the task does not naturally improve by scaling up only (e.g., year parity).

实验结果



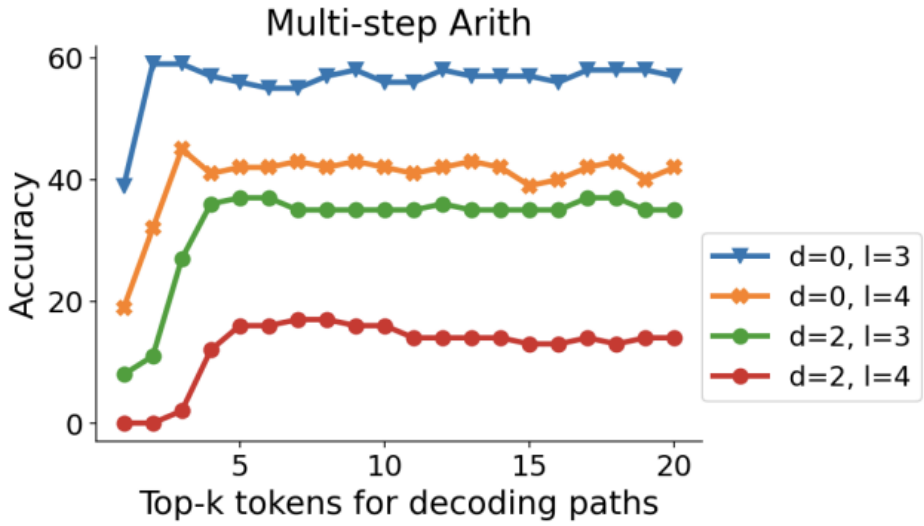


Figure 5 | The effect of k on reasoning accuracy w.r.t. PaLM-2 model scales and task difficulty.

实验结果

	Coin Flip			Web of lies			Multi-step Arithmetic				Sports Und.	Object Count
	2	3	4	3	4	5	d_0, l_3	d_0, l_4	d_2, l_3	d_2, l_4		
Greedy	70.0	53.0	48.0	76.0	58.0	53.6	39.0	19.0	8.0	0.0	58.8	36.0
CoT-decoding	**94.0**	**57.0**	**55.0**	**87.0**	**63.0**	**57.6**	**56.0**	**42.0**	**35.0**	**16.0**	58.0	**39.2**

Table 6 | The model's intrinsic reasoning ability varies depending on the task difficulty levels.

尽管CoT解码在几乎所有任务上都帮助引出了更好的推理，但是在不同任务难度级别上差异显著：任务越简单，找到正确推理路径的可能性越大。作者查看了模型的top-k解码路径，发现当解答只涉及最多1或2步知识操作时，模型可以生成正确的CoT路径；而当步骤增加到3步或更多时，模型开始难以生成正确的CoT路径。见图5（右侧），随着任务复杂性增加（较高的d和l值），模型的准确率仅在更大的k值下有所改善。这一现象表明，随着任务变得更合成化，正确的CoT路径变得更难以找到。即语言模型的表现受其训练分布的影响很大。

实验结果

总结

1. **CoT解码揭示了模型在推理中的内在脆弱性:**模型可以生成逐步模拟过程的CoT路径，但在任务复杂性增加时容易丢失状态，这揭示了模型在进行准确状态跟踪方面的内在脆弱性。在多步算术任务中，作者观察到模型在CoT解码路径中往往从左到右依次进行计算，而不是遵循正确的数学运算顺序。这些观察指出了未来改进模型的方向。。

2.在多模态内容理解是不是可以考虑下解码的方式，而不是通过数据集微调或者模型结构一些修改，这种方法在比赛中可不可以考虑。

Thank for your attention !