Cumulative Reasoning With Large Language Models

Abstract

即使大语言模型功能强大而且丰富, 他们在解决高度复杂问题上仍然不太行. 这是因为解决负载问题需要自由的思考, 但这在训练过程中很少被指导. 在本文中, 我们提出一种新的方法被称为**累计推理/Cumulative Reasoning**, 以累计/迭代的方式来模仿人类思考的过程. 通过讲任务分解为多个小的成分，CR可以讲问题解决过程流水线化，使得其更加的可管理和高效. CR在FOLIOwiki数据集和24点游戏上比其他方法性能显著的好. 在MATH数据集上, 58.0%准确率, 新的SOTA.

Introduction

LLMs在困难推理问题上比较差.
我们的认知过程包括两个不同的系统, 系统1是快速的, 本能的, 情感的; 系统2是慢速的, 深思熟虑的, 逻辑的. LLMs更像是系统1.
现有方法, CoT, ToT等等. 然而这些方法都没有一个存储中间结果的地方, 假设所有的思维形成一个链or树, 这并没有完全捕捉到人类的思维过程.
我们提出了CR, cumulative learning, 对思考具有更加一般的刻画. CR使用三种不同的LLM, proposer, verifier, reporter. Proposer提出潜在可能的命题, 通过一个or多个verifier验证, 而reporter决定何时停止并且报告solution.
CR显著增强了语言模型在处理复杂问题方面的能力.
实验方面, 首先使用 FOLIO wiki / AutoTNLI, 分别涉及一阶逻辑和高阶逻辑. 然后24点游戏. 然后MATH数据集

Example of Logic

1. All monkeys are mammals: $\forall x(\text{Monkey}(x) \Rightarrow \text{Mammals}(x))$.
2. An animal is either a monkey or a bird: $\forall x(\text{Animal}(x) \Rightarrow (\text{Monkey}(x) \vee \text{Bird}(x)))$.
3. All birds fly: $\forall x(\text{Bird}(x) \Rightarrow \text{Fly}(x))$.
4. If something can fly, then it has wings: $\forall x(\text{Fly}(x) \Rightarrow \text{Wings}(x))$.
5. Rock is not a mammal, but Rock is an animal: $\neg\text{Mammal}(\text{Rock}) \wedge \text{Animal}(\text{Rock})$.

The question is: does Rock have wings? We have the following derivations:

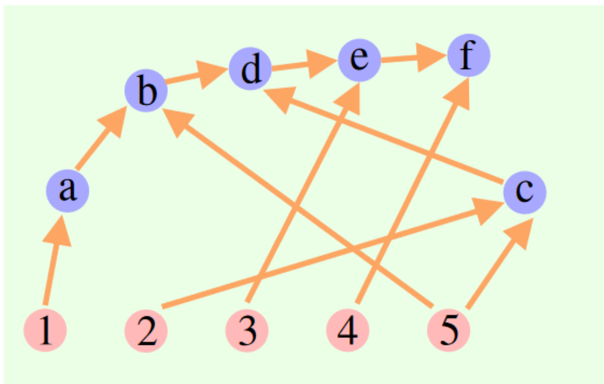


Figure 1: Illustration of our logical derivation

a. The contrapositive of (1) is: $\forall x(\neg\text{Mammals}(x) \Rightarrow \neg\text{Monkey}(x))$.
b. (a) and (5) $\Rightarrow \neg\text{Monkey}(\text{Rock}) \wedge \text{Animal}(\text{Rock})$.
c. (2) and (5) $\Rightarrow (\text{Monkey}(\text{Rock}) \vee \text{Bird}(\text{Rock}))$
d. (b) and (c) $\Rightarrow \text{Bird}(\text{Rock})$.
e. (3) and (d) $\Rightarrow \text{Fly}(\text{Rock})$.
f. (4) and (e) $\Rightarrow \text{Wings}(\text{Rock})$.

这种推理过程不是CoT/ToT, 而是图

> 对此表示反对, 依然是CoT啊, 推理过程就是一步一步的.
> 不过, 如果推理步骤出错了, 如何回退是一个问题.
> Remark: CR就是能够回退, 这一点确实比CoT强很多.

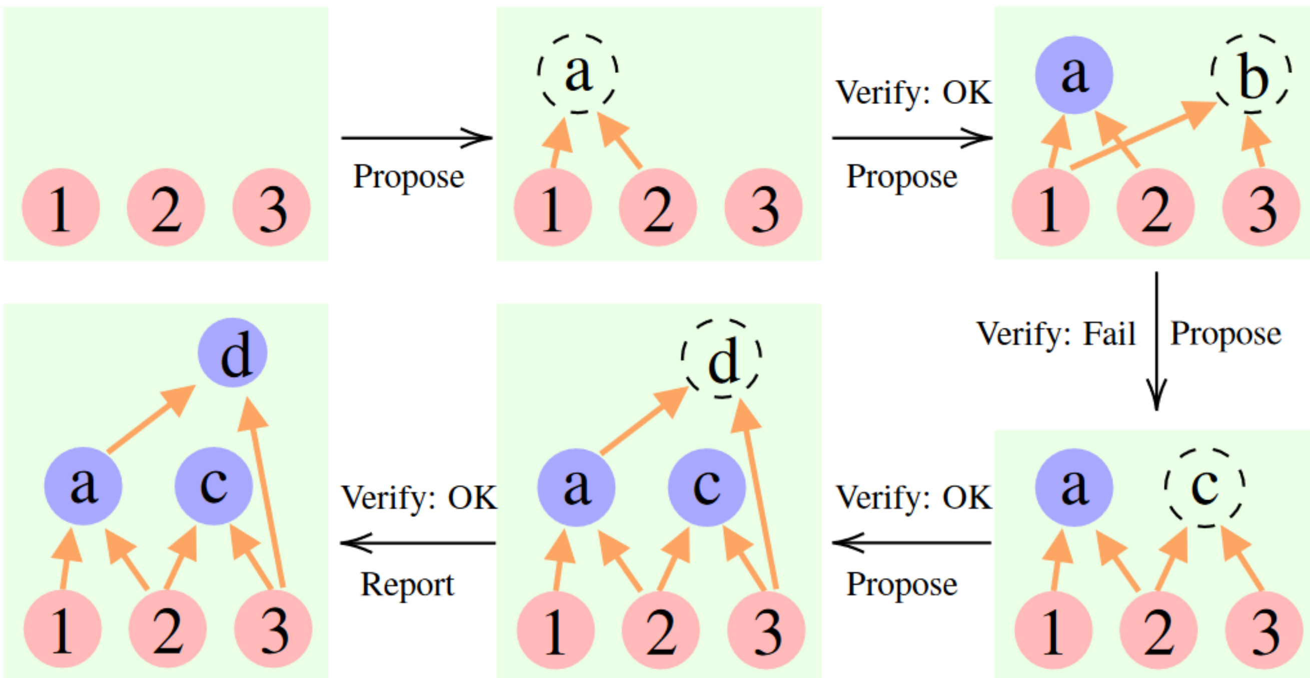
Method

Cumulative Reasoning

三个类型的LLMs:

- proposer: 基于当前的context, 提出下一步;

- verifier(s): 这个模型会仔细审查proposer提出的步骤的准确性, 如果是正确的, 则会加入到context中;
- reporter: 通过评估当前的条件是否能直接得到最终的答案, 决定推理过程是否结束



与CoT/ToT的比较

CR明显是对CoT的一种泛化, 如果没有verifier, proposer一直提出下一步, 指导结束. 但由于CR中, 整体的思维过程可以是一个有向无环图, 所以可以解决更加复杂的问题.
ToT和CR看起来很像, 但是CR会将历史上所有的正确的推理结果存放在内存中.

> 所以CR到底是DFS还是BFS呢, BFS必然是不行的吧, 因为分支数目过多. 但是DFS也只有当前这一条可能正确的推理路径.

Experiments

Setting

- GPT3.5-turbo
- GPT4
- LLaMA-13B
- LLaMA-65B

  CR中Proposer / Verifier / Reporter使用相同的LLM, 不同的prompt. 未来可以考虑特定任务语料上训练的Proposer, 使用形式逻辑系统辅助的Verifier

FOLIO wiki

Table 1: Results for various reasoning approaches on FOLIO-wiki dataset.

Model	Method	Acc. ↑ (%)	Error ↓ (%)
-	[Random]	33.33	66.67
LLaMA-13B	Direct	44.75	55.25
	CoT	49.06 (+4.31)	50.94 (-4.31)
	CoT-SC (k = 16)	52.43 (+7.68)	47.57 (-7.68)
	CR (ours, n = 2)	**53.37** (+8.62)	**46.63** (-8.62)
LLaMA-65B	Direct	67.42	32.58
	CoT	67.42 (+0.00)	32.58 (-0.00)
	CoT-SC (k = 16)	70.79 (+3.37)	29.21 (-3.37)
	CR (ours, n = 2)	**72.10** (**+4.68**)	**27.90** (**-4.68**)
GPT-3.5-turbo	Direct	62.92	37.08
	CoT	64.61 (+1.69)	35.39 (-1.69)
	CoT-SC (k = 16)	63.33 (+0.41)	36.67 (-0.41)
	CR (ours, n = 2)	**73.03** (**+10.11**)	**26.97** (**-10.11**)
GPT-4	Direct	80.52	19.48
	CoT	84.46 (+3.94)	15.54 (-3.94)
	CoT-SC (k = 16)	85.02 (+4.50)	14.98 (-4.50)
	CR (ours, n = 2)	**87.45** (**+6.93**)	**12.55** (**-6.93**)

FOLIO wiki curated

Auto TNLI

Tabular Natural Language Inference. 可以视为高阶逻辑推理数据集.

Method	Acc. ↑ (%)	# Avg. visited states ↓
Direct	7.3	1
CoT	4.0	1
CoT-SC (k = 100)	9.0	100
Direct (best of 100)	33	100
CoT (best of 100)	49	100
ToT (b = 5)	74	61.72
CR (ours, b = 1)	84 (+10)	**11.68** (**-50.04**)
CR (ours, b = 2)	94 (+20)	13.70 (-48.02)
CR (ours, b = 3)	97 (+23)	14.25 (-47.47)
CR (ours, b = 4)	97 (+23)	14.77 (-46.95)
CR (ours, b = 5)	**98** (**+24**)	14.86 (-46.86)

24点

MATH

Table 5: Comparative performance on the MATH dataset using GPT-4. We adopted a default temperature setting of $t = 0.0$, consistent with prior research settings (greedy decoding). PHP denotes the application of the progressive-hint prompting. "Iters" represents the average number of LLM interactions, and **Overall** reflects the overall results across MATH subtopics.

	w/ PHP	MATH Dataset (* denotes using 500 test examples subset following Lightman et al. (2023))							
		InterAlgebra	Precalculus	Geometry	NumTheory	Probability	PreAlgebra	Algebra	**Overall**
CoT (OpenAI, 2023)	✗	-	-	-	-	-	-	-	42.50
Complex CoT, 8-shot (Zheng et al., 2023)	✗	23.4	26.7	36.5	49.6	53.1	71.6	70.8	50.36
	✓	26.3	29.8	41.9	55.7	56.3	73.8	74.3	53.90
	(Iters)	3.2414	3.2435	3.2233	3.1740	2.8122	2.3226	2.4726	2.8494
Complex CoT* (repro., 8-shot)	✗	29.9	33.9	34.1	46.8	47.4	62.1	70.7	48.80
	✓	28.9	30.4	43.9	53.2	50.0	68.5	84.1	53.80
	(Iters)	2.7629	2.4643	2.7805	2.7581	2.4474	2.3780	2.5484	2.59
CR* (**ours**, 4-shot)	✗	28.9 (-1.0)	30.4 (-3.5)	39.0 (+4.9)	54.8 (+8.0)	57.9 (+10.5)	71.8 (+9.7)	79.3 (+8.6)	**54.20** (+5.40)
	✓	**32.0** (+3.1)	**35.7** (+5.3)	**43.9** (+0.0)	**59.7** (+6.5)	**63.2** (+13.2)	**71.8** (+3.3)	**86.6** (+2.5)	**58.00** (+4.20)
	(Iters)	2.6598	2.4821	2.5122	2.2903	2.2105	2.2195	2.3548	**2.40** (-0.19)

Table 6: Comparative performance on the MATH dataset using GPT-4 for different difficulty levels.

	w/ PHP	MATH Dataset (* denotes using 500 test examples subset)					
		Level 5	Level 4	Level 3	Level 2	Level 1	**Overall**
CoT (OpenAI, 2023)	✗	-	-	-	-	-	42.50
Complex CoT* (repro., 8-shot)	✗	22.4	38.3	62.9	72.2	79.1	48.80
	✓	23.9	43.8	63.8	86.7	83.7	53.80
CR* (**ours**, 4-shot)	✗	**32.1** (+9.7)	43.0 (+4.7)	62.9 (+0.0)	78.9 (+6.7)	83.7 (+4.6)	**54.20** (+5.40)
	✓	27.3 (+3.4)	**50.0** (+6.2)	**70.9** (+7.1)	**86.7** (+0.0)	**90.7** (+7.0)	**58.00** (+4.20)

分别比较了CoT / Complex CoT / CR 以及 w / wo PHP(progressive hint prompt)