

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Рецензия

Пяцько Даниил

Recap

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

ЭВОЛЮЦИЯ

Version Control

V5 → V6. Fixed minor typo in Figure 3.

V4 → V5. Added Codex and UL2 results. Small changes to writing and style of paper.

V3 → V4. Fixed typo in Figure 3 and added a couple citations.

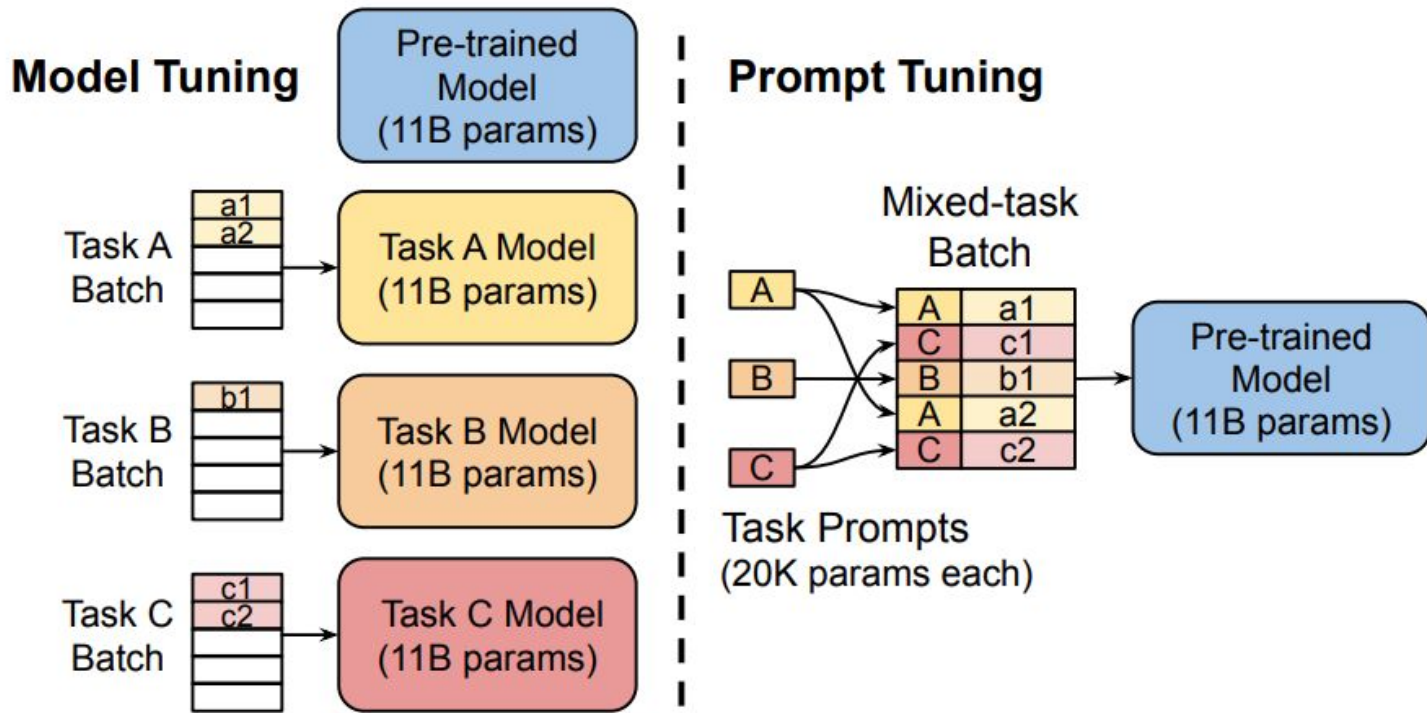
V2 → V3. Added GPT-3 results. Added SVAMP and AQuA eval datasets for math. Added SayCan eval for commonsense. Added Extended Related Work section (Appendix C). Added ablations for Commonsense and Symbolic Reasoning (Table 7). Added FAQ section (Appendix A). Added raw results in Appendix B.

V1 → V2. Added PaLM results (V1 only had LaMDA).

Авторы

- 9 авторов, все из Google Research
- Jason Wei, Xuezhi Wang и Denny Zhou после продолжили заниматься CoT
- Также эти трое соавторы статьи *PaLM: Scaling Language Modeling with Pathways* (Chowdhery et. al)

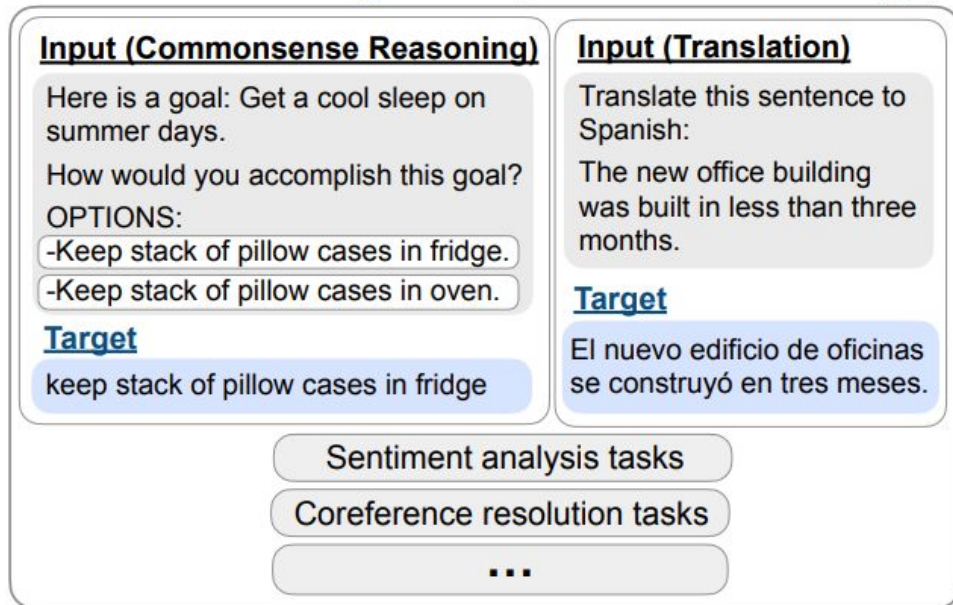
Предшественники 1



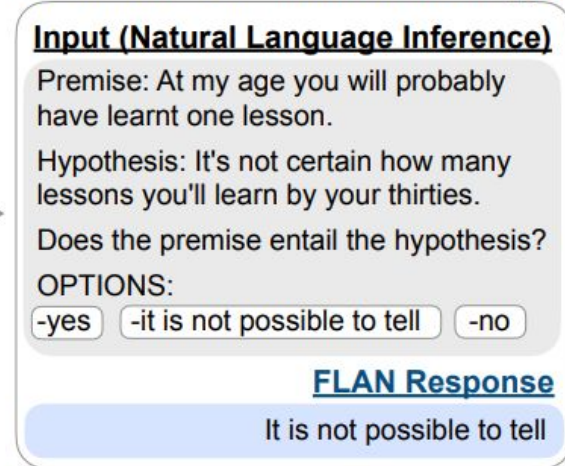
The Power of Scale for Parameter-Efficient Prompt Tuning (Lester et. al)

Предшественники 2

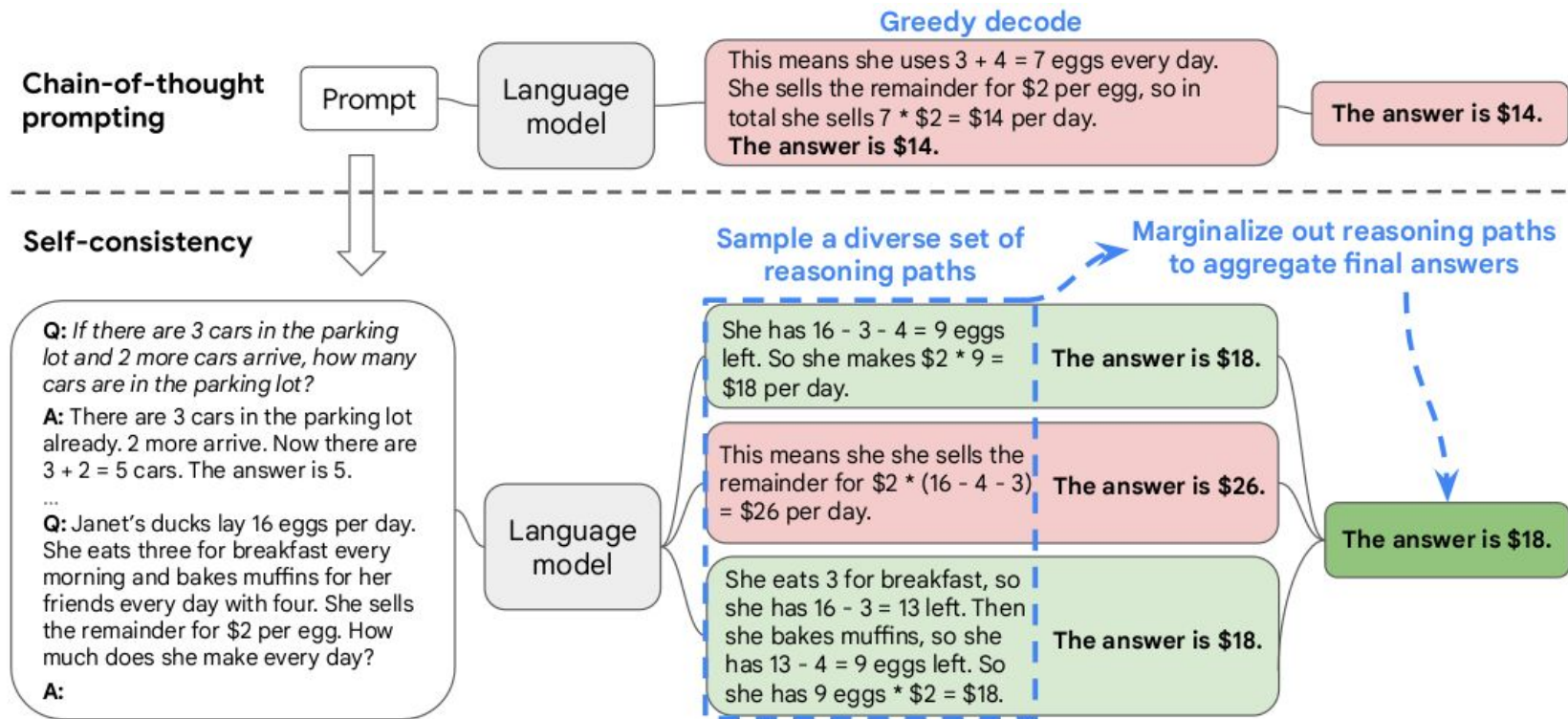
Finetune on many tasks (“instruction-tuning”)



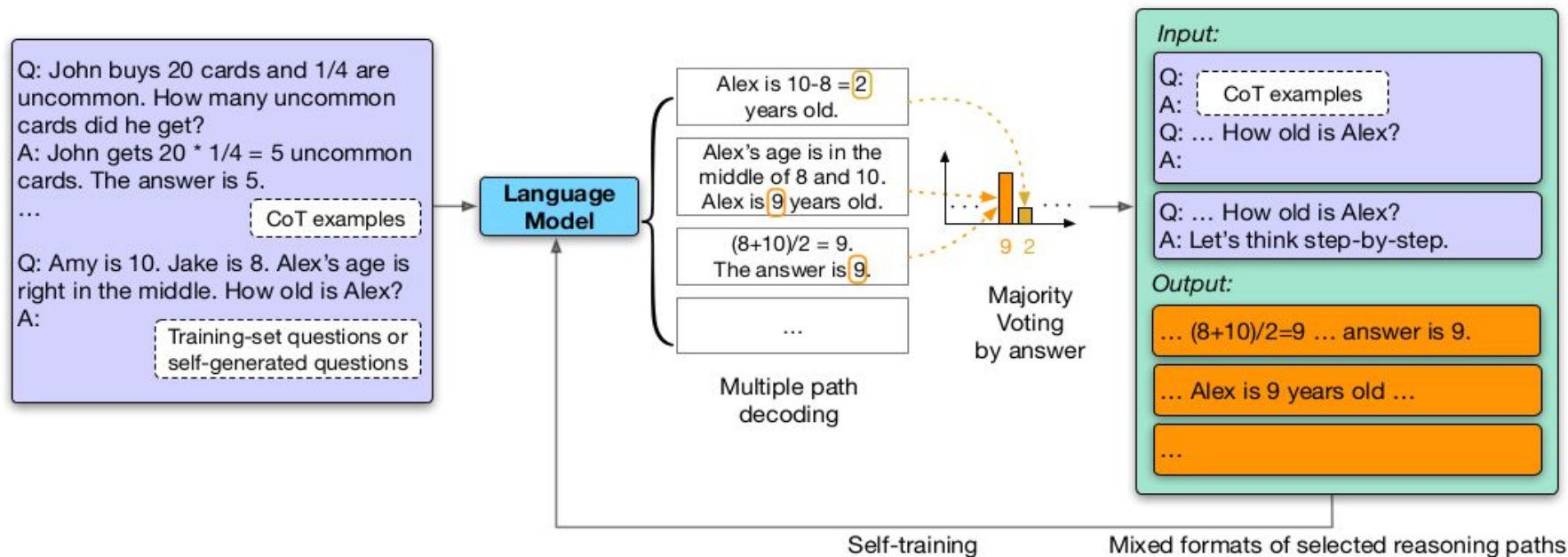
Inference on unseen task type



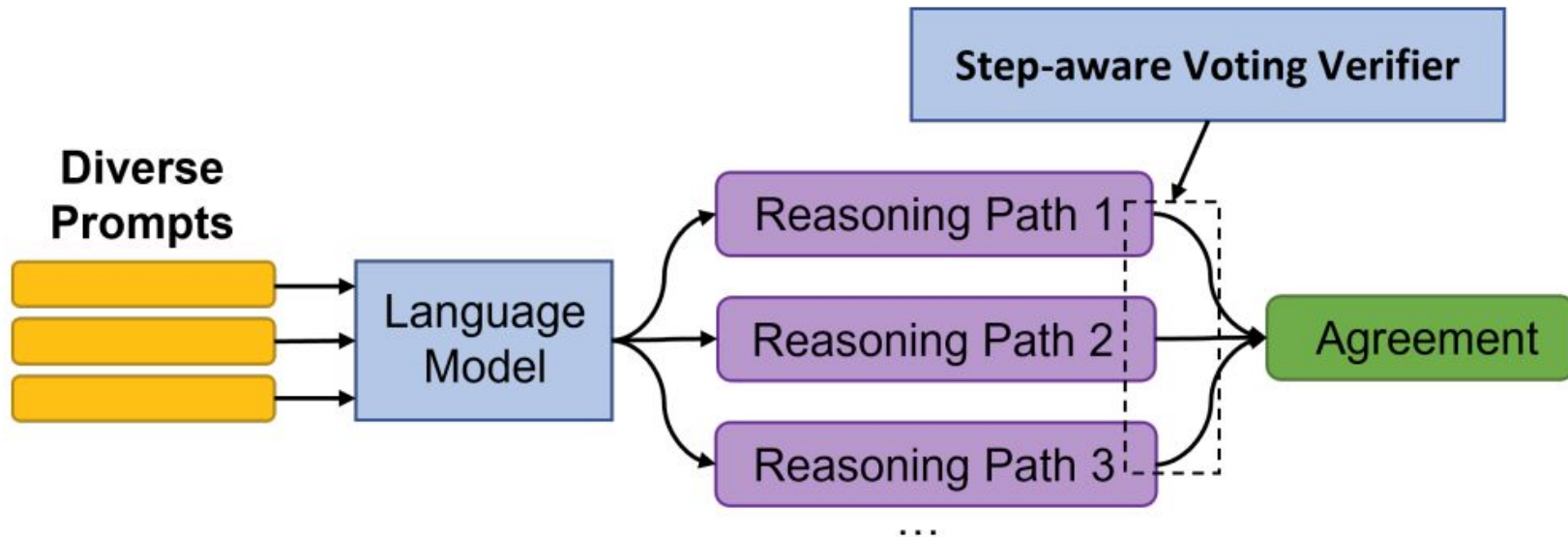
Продолжение 1



Продолжение 2



Продолжение 3



Продолжение 4

Chain-of-Thought (Wei et al., 2022)

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 tennis balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The bakers at the Beverly Hills Bakery baked 200 loaves of bread on Monday morning. They sold 93 loaves in the morning and 39 loaves in the afternoon. A grocery store returned 6 unsold loaves. How many loaves of bread did they have left?

Model Output

A: The bakers started with 200 loaves. They sold 93 in the morning and 39 in the afternoon. So they sold $93 + 39 = 132$ loaves. The grocery store returned 6 loaves. So they had $200 - 132 - 6 = 62$ loaves left. The answer is 62.

Program-aided Language models (this work)

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 tennis balls.
`tennis_balls = 5`
2 cans of 3 tennis balls each is
`bought_balls = 2 * 3`
tennis balls. The answer is
`answer = tennis_balls + bought_balls`

Q: The bakers at the Beverly Hills Bakery baked 200 loaves of bread on Monday morning. They sold 93 loaves in the morning and 39 loaves in the afternoon. A grocery store returned 6 unsold loaves. How many loaves of bread did they have left?

Model Output

A: The bakers started with 200 loaves
`loaves_baked = 200`
They sold 93 in the morning and 39 in the afternoon
`loaves_sold_morning = 93`
`loaves_sold_afternoon = 39`
The grocery store returned 6 loaves.
`loaves_returned = 6`
The answer is
`answer = loaves_baked - loaves_sold_morning`
`- loaves_sold_afternoon + loaves_returned`

```
>>> print(answer)  
74
```



GSM8K Топ

Продолжение 3

1	DIVERSE 175B (8-shot)	83.2	175	✓	On the Advance of Making Language Models Better Reasoners		2022	<div>few-shot</div>
---	---------------------------------	------	-----	---	---	---	------	---------------------

Продолжение 2

2	PaLM 540B (Self Improvement, Self Consistency)	82.1		×	Large Language Models Can Self-Improve		2022	
---	--	------	--	---	--	---	------	--

3	Minerva 540B-maj1@k (8-shot)	78.5	540	✓	Solving Quantitative Reasoning Problems with Language Models		2022	<div>few-shot</div>
---	--	------	-----	---	--	---	------	---------------------

Продолжение 1

4	PaLM 540B maj1@40 (8-shot)	74.4	540	✓	Self-Consistency Improves Chain of Thought Reasoning in Language Models		2022	<div>few-shot</div>
---	--------------------------------------	------	-----	---	---	---	------	---------------------

5	PaLM 540B (Self Consistency)	74.4		×	Large Language Models Can Self-Improve		2022	
---	--	------	--	---	--	---	------	--

6	PaLM 540B (Self Improvement, CoT Prompting)	73.5		×	Large Language Models Can Self-Improve		2022	
---	---	------	--	---	--	---	------	--

Сильные и слабые стороны

Сильные стороны

1. Актуальный способ улучшить качество, так как LM идут в сторону увеличения параметров, а метод как раз хорошо работает для больших моделей.
2. Корректные эксперименты, сравнение с SOTA моделями.
3. Много продолжений, которые дают SOTA результат на GSM8K
4. Простой метод, ничего дообучать не надо.

Слабые стороны

1. Недостаточно исследован prompting и влияние на результат
2. Метод дает профит только на больших моделях, которые доступны далеко не всем.

Возможные улучшения

1. Попытайтесь увеличить качество на маленьких моделях
2. Эксперименты с prompting'ом. В статье сравниваются промпты, написанные двумя разными людьми. Хотя и написанные разными людьми, промпты бывают весьма похожи. Хочется какую-то семантическую разницу.
3. (Не по статье, а в целом) Кажется, что можно скрестить второе и третье продолжение, то есть дообучить на сгенерированных же ею данных и сделать более хитрый выбор ответа, чем выбор того, кто встретился наибольшее число раз. На такой модели есть шанс получить еще больше на GSM8k.