Meeting Note

Author: Henry Lin
Date: 2024/09/02 – 2024/09/06

Weekly Summary
Chain-of-Thought Prompting Elicits Reasoning in Large Language Models [1].

Plan for Next Week
Attention Is All You Need (arXiv: 1706.03762)

Details
Although fine-tuning and scaling up the size of large language models (LLMs) can improve the performance on various tasks, these are not efficient and show limited performance on challenging tasks. To address these problems, Wei et al. [1] propose a simple yet powerful method called chain-of-thought to enhance the reasoning ability of LLMs using prompting techniques. In this study, the authors experiment three models (PaLM 540B [2], GPT-3 [3], and LaMDA [4]) across three tasks: arithmetic, commonsense, and symbolic reasoning, which are challenging for LLMs.

The proposed chain-of-thought is mainly inspired by two ideas: (1) techniques for arithmetic reasoning can benefit from generating natural language rationales leading to the answer, which means, when solving arithmetic problems, explaining the reasoning process step-by-step with natural language can help the model achieve correct answer; (2) LLMs shows potential of in-context few-shot learning via prompting, that is, instead of fin-tuning, prompting LLMs with a few simple input-output examples has been successful for a range of tasks. In addition, the prompting-only approach is valuable because it is cost-effective (e.g. large training dataset is not required) and preserves the generality of LLMs, as the fine-tuning is unnecessary.

By combining the strengths of the above methods, the authors give a prompting template: "input, chain-of-thought, output" where chain-of-thought is a coherent series of intermediate reasoning steps that lead to the final answer, to achieve the simple objective of this work: giving LLMs the ability of generating chain of thought. And some properties of this approach have been introduced: (1) chain-of-thought allows LLMs to decompose a problem into multiple steps; (2) it provides interpretability which can help debugging and improving the LLMs; (3) it shows potential application to any task that can be solved via natural language; (4) it can be used on any trained LLMs directly.

For the experiments of solving arithmetic problems, a set of eight question-answer (QA) pairs with chain-of-thought prompting as few-shot exemplar is manually composed as follow, which is used for all benchmarks:

PROMPT FOR MATH WORD PROBLEMS

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

A: There are originally 3 cars. 2 more cars arrive. 3 + 2 = 5. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?

A: Originally, Leah had 32 chocolates. Her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?

A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he gave Denny 20 - 12 = 8. The answer is 8.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?

A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. 5 + 4 = 9. The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: There were originally 9 computers. For each of 4 days, 5 more computers were added. So 5 * 4 = 20 computers were added. 9 + 20 is 29. The answer is 29.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?

A: Michael started with 58 golf balls. After losing 23 on tuesday, he had 58 - 23 = 35. After losing 2 more, he had 35 - 2 = 33 golf balls. The answer is 33.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?

A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 5 x 3 = 15 dollars. So she has 23 - 15 dollars left. 23 - 15 is 8. The answer is 8.

---

where the highlighted sentences are chain-of-thought promptings. The experimental result in Figure 1 shows two phenomena: (1) chain-of-thought improves the performance on large models with over 100B parameters, but has no significant impact on small models; (2) it has larger performance gains for complicated tasks such as GSM8K [5]. On the easiest benchmark MAWPS [6], the performances with and without chain-of-thought prompting are similar.

To understand why chain-of-thought works, the authors randomly pick 50 generated chains of thought for each correctly answered examples and incorrectly answered ones, they found that 46% of the chains of thought are correct (or with only minor errors), and 54% had major errors in sematic understanding and logical coherence [Re 1].
Furthermore, they also analyzed the results generated by PaLM 62B and PaLM 540B to assess the performance gains in larger models, finding that 540B significantly improved semantic understanding and reduced one-step omission errors present in the 62B model.

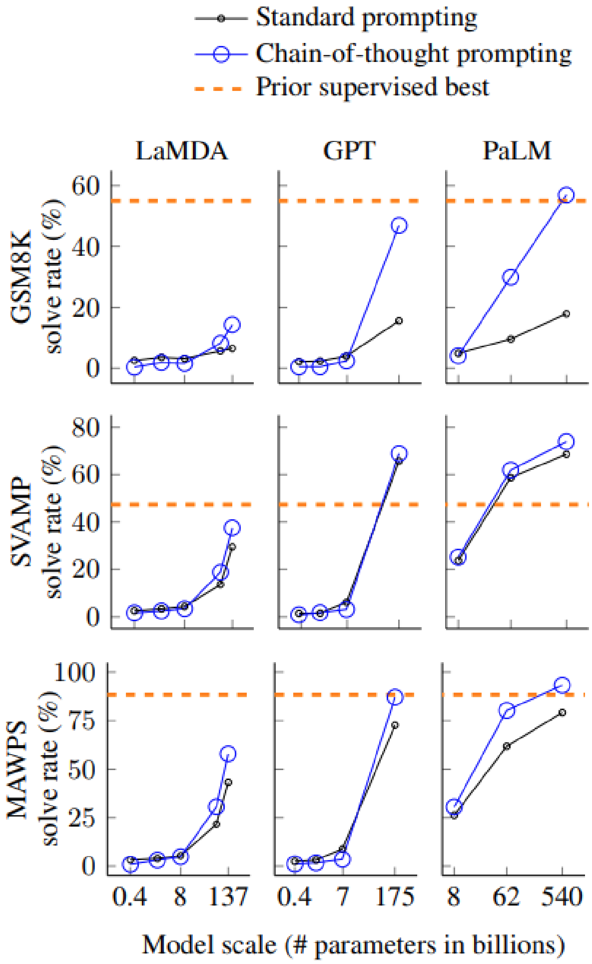

Figure 1. The experimental results of arithmetic problems of three LLMs.

To observe the benefits of chain-of-thought, the authors perform the ablation study with three variations on chain-of-thought: (1) equation only, where the model is asked to only output the mathematical equation before giving the answer. Notably, the experiments shows that this method does improve the performance for easy questions which only require very few steps to solve; (2) variable compute only, where the model is asked to output the equation before giving answer, and replace all the characters in the equation with dots

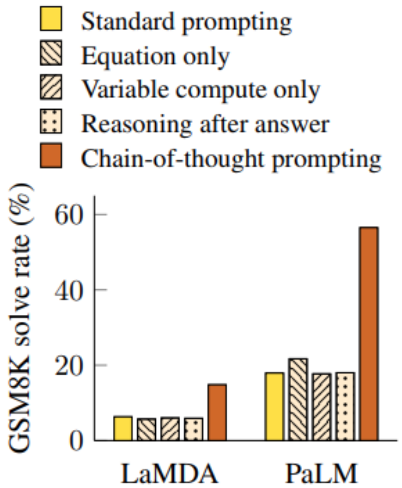

Figure 2. Ablation studies of LaMDA 137B and PaLM 450B.

(…) [Re 2]; (3) chain-of-thought after answering, where the model is asked to output the chain-of-thought after answering the questions. As shown in Figure 2, none of the variations of chain-of-thought significantly improve the performance of LLMs, demonstrating the effectiveness of using chain-of-thought prompting.

Figure 3 shows the results of LaMDA 137B on GSM8K and MAWPS using different annotators and exemplars. As we can see, all the different sets of chain-of-thought prompting outperform the standard prompting, this result shows the robustness of this proposed prompting technique.

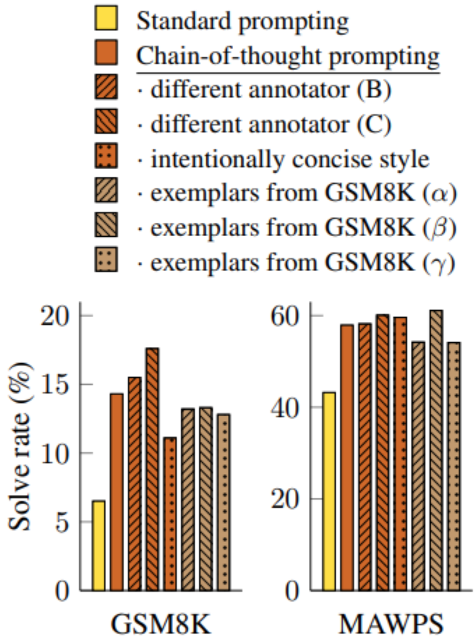

Figure 3. Robustness testing of chain-of-thought using LaMDA 137B.

The authors also did experiments on the other two tasks (commonsense and symbolic reasoning), and got the similar results to arithmetic problems [Re 3]. (Due to the similarity in methodologies and results, this section will be skipped.)

Table 1. The results of ablation study and robustness testing.

	GSM8K	SVAMP	ASDiv	MAWPS
Standard prompting	6.5 ±0.4	29.5 ±0.6	40.1 ±0.6	43.2 ±0.9
Chain of thought prompting	14.3 ±0.4	36.7 ±0.4	46.6 ±0.7	57.9 ±1.5
Ablations				
· equation only	5.4 ±0.2	35.1 ±0.4	45.9 ±0.6	50.1 ±1.0
· variable compute only	6.4 ±0.3	28.0 ±0.6	39.4 ±0.4	41.3 ±1.1
· reasoning after answer	6.1 ±0.4	30.7 ±0.9	38.6 ±0.6	43.6 ±1.0
Robustness				
· different annotator (B)	15.5 ±0.6	35.2 ±0.4	46.5 ±0.4	58.2 ±1.0
· different annotator (C)	17.6 ±1.0	37.5 ±2.0	48.7 ±0.7	60.1 ±2.0
· intentionally concise style	11.1 ±0.3	38.7 ±0.8	48.0 ±0.3	59.6 ±0.7
· exemplars from GSM8K (α)	12.6 ±0.6	32.8 ±1.1	44.1 ±0.9	53.9 ±1.1
· exemplars from GSM8K (β)	12.7 ±0.5	34.8 ±1.1	46.9 ±0.6	60.9 ±0.8
· exemplars from GSM8K (γ)	12.6 ±0.7	35.6 ±0.5	44.4 ±2.6	54.2 ±4.7

Remarks

[Re 1]

How were the 46% and 54% determined? The authors mentioned that in the 50 correctly answered examples, the chains of thought were also logically correct except two of them arrived the correct answer coincidently, which means, in these 50 examples, 48 of the chains of thought are correct, 2 are incorrect.

[Re 2]

This variant seems to be similar to the variant of equation only.

[Re 3]

Figure 4 and 5 show significant improvements of chain-of-thought.

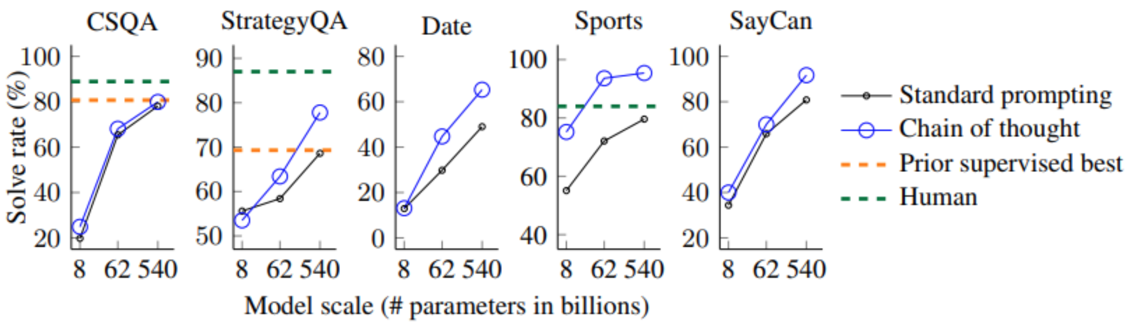

Figure 4. The experimental results on the task of commonsense using PaLM.

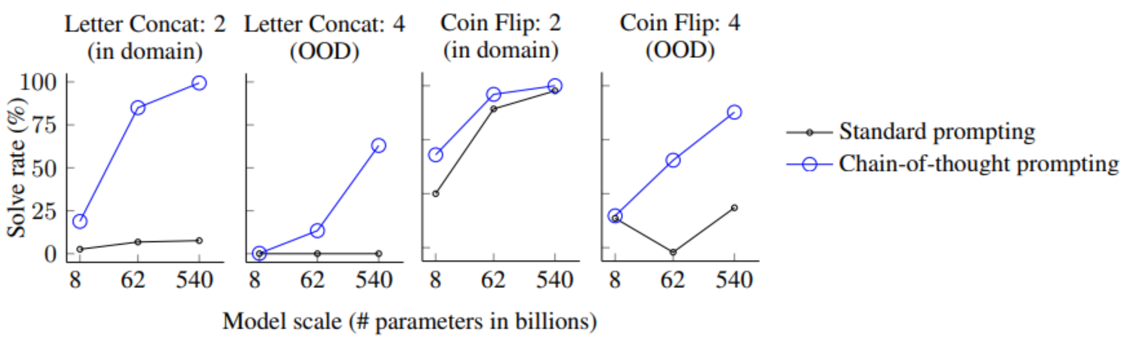

Figure 5. The experimental results on the task of symbolic reasoning using PaLM.

[Re 4]

We also did a simple test on modern LLMs. Figure 6 shows that the reasoning abilities of LLMs have largely improved in recent years. The models can provide the correct answers of simple questions even when asked to response directly.

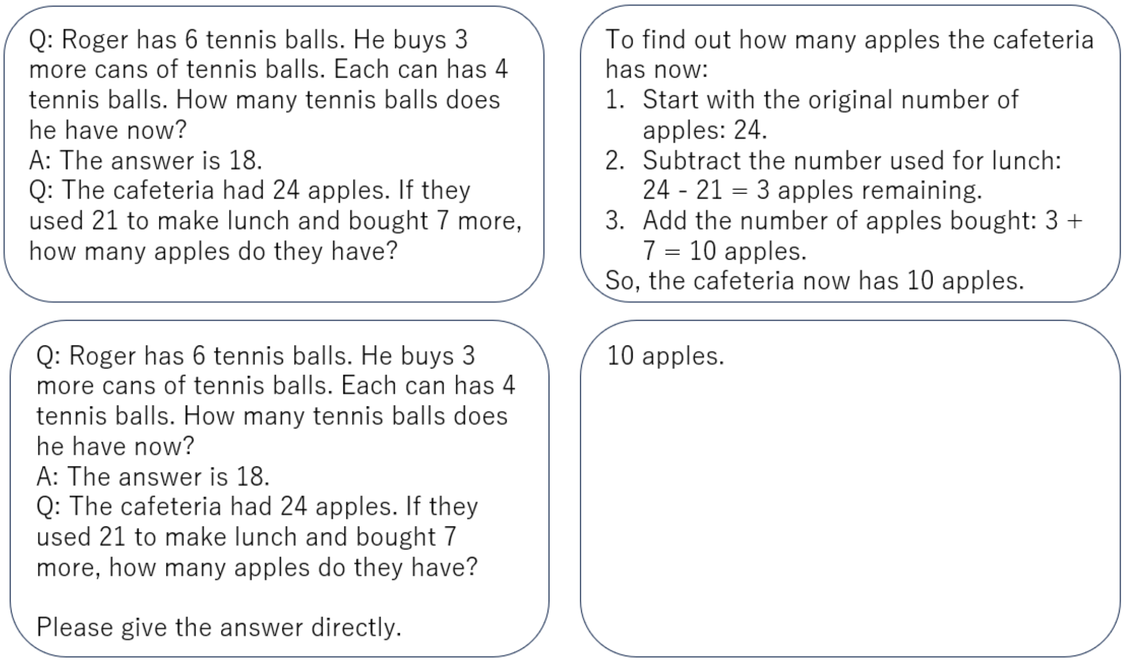

Figure 6. The squares on the right side are the arithmetic problems, and the squares on the left side are the corresponding outputs of GPT-4o.

References

[1] WEI, Jason, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 2022, 35: 24824-24837.

[2] CHOWDHERY, Aakanksha, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 2023, 24.240: 1-113.

[3] BROWN, Tom, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 2020, 33: 1877-1901.

[4] THOPPILAN, Romal, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[5] COBBE, Karl, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[6] KONCEL-KEDZIORSKI, Rik, et al. MAWPS: A math word problem repository. In: *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*. 2016. p. 1152-1157.

Comments

N/A