1주차 발표 준비 (백성은)

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

https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

논문 선정 이유 : prompt engineering에 대한 가장 기본적인 기법을 다룬 논문이라서.

Pre-study

Language Models are Few-Shot Learners

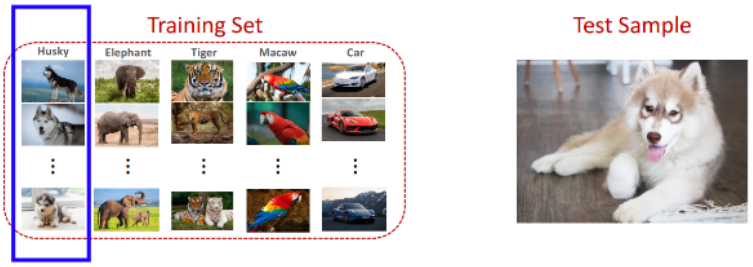
▼ in-context learning이란?

prompt의 내용만으로 task를 수행할 수 있도록 작업 → 즉, prompt의 맥락 (in-context)을 파악하고 이에 대한 답변을 생성하도록 만듬.

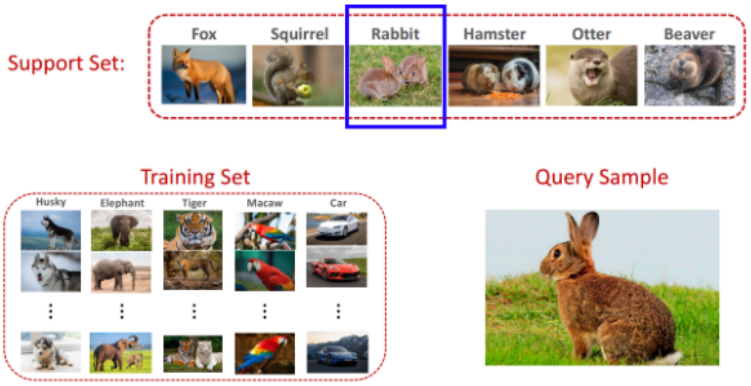
zero-shot / one-shot / few-shot : 기본적으로 train set에 포함되지 않은 label, class에 대한 예측을 할 수 있도록 학습하는 방법

Few shot learning vs Supervised learning

- **Supervised learning** : Test image (Query image) 의 클래스가 Training set에 있음!! -> 학습에 강아지 사진을 주고 강아지를 잘 학습했는 지 묻는 것!



- **Few shot learning** : **Training set에 없는 클래스를 맞추는 문제**(Zero, One, Few shot)



- zero-shot : 어떤 추가적인 데이터 없이, 기존 학습에 사용된 데이터에서 새로운 class를 인식하고 분류하는 학습 방법
  - 이미 알고 있는 class 간에 관계에 대한 사전 지식을 활용하여 새로운 class를 예측
  - **class에 해당되는 추가 데이터 대신, class에 대한 (혹은 task에 대한) 설명을 제시함으로써 학습 가능**
    - text, attribute, class similarity 등과 같은 추가 정보들을 제시
  - semantic embedding과 feature (attribute) 기반 학습으로 training에 활용 가능
    - semantic embedding은 단어 or 문장을 vector space로 mapping하는 방법 (word2vec)
- one-shot / few shot : **class에 해당 되는 데이터를 제공하여 새로운 class에 대해 학습**
  - one-shot : 제시되는 데이터가 1개일 경우
    - 개와 고양이를 구분하는 모델에서 사자를 예측하게 할 때, 사자 이미지를 1장 주는 것

    - few-shot : 제시되는 데이터가 여러 개일 경우 (일반적으로 10개 미만)

공부하다가 드는 의문

- prompting과 prompt engineering은 다른 것인가?
    - prompting : 언어 모델에 특정 prompt를 입력하여 원하는 출력을 얻어내는 과정
    - prompt engineering : 원하는 출력을 얻을 수 있는 최적의 prompt를 설계하고 조정하는 과정
    - 쉽게 말해서, 우리가 요리하는 과정이 prompting, 그 과정에서 여러 레시피를 참고하여 재료를 추가하거나 빼는 등의 작업이 prompt engineering
- CoT는 prompt engineering의 일부가 아닌가?
    - 애매한 것 같음. prompting 방식 중 하나인 것 같고, prompting을 정한 후에 prompt engineering을 할 수 있는 가능성도 존재하는 것 같음.
- few-shot learning은 그럼 prompt engineering인가?
    - 의미가 조금 다른 것 같은데, few-shot learning은 모델 학습 능력 자체를 활용하여 새로운 task를 수행하게 만드는 반면, prompt engineering은 학습 능력을 효과적으로 활용하기 위한 방법론인 것 같음.

Abstract

- chain of thought를 생성하는 것이 어떻게 LLM이 복잡한 추론 문제에 대한 능력을 향상시킬 수 있는 지에 대해 탐구
- 특히, chain of thought prompting이라고 불리는 간단한 방법을 통해 LLM이 어떻게 추론 능력을 갖게 되는 지에 대해서 알리고자 함.
    - chain of thought prompting : chain of thought demonstrations을 prompting 안에서 examplars로써 제공하는 것
- arithmetic, commonsense, symbolic reasoning task에서 뛰어난 성능을 보임

Introduction

- 최근 NLP 분야는 많은 발전을 이룩했지만, arithmetic (산술), commonsense, symbolic reasoning과 같은 분야에서는 높은 performance를 보여주지 못한다.
- LLM의 reasoning ability를 간단한 2가지 아이디어로부터 motivated된 방법으로 탐구
  - arithmetic reasoning은 최종 답변을 유도하기 위한 natural language를 생성함으로써 이득을 얻을 수 있다.
    - 이전 연구는 모델을 처음부터 훈련하거나 fine-tuning, 혹은 자연어 대신 formal language를 사용하는 neuro-symbolic methods를 통해 중간 단계의 자연어를 생성할 수 있도록 유도
  - prompting을 통해 in-context few shot learning
    - 즉, 별도의 fine-tuning없이 단지 모델에 몇 가지 input-output에 대한 예시를 제공하는, 'prompt' 함으로써 task를 수행
- 하지만 이 2가지 방법 모두 한계가 존재
  - from scratch로 훈련하거나 fine-tuning하는 것은 cost가 높음
  - 'prompt'는 일부 추론 능력이 필요한 task에서는 효과적인 모습을 보여주지 못함
- **이 논문에서는 2가지 방법의 강점들을 모아 한계 점을 해결하고자 시도**
- 'input + chain of thought + output' 으로 구성된 prompt를 제공
  - chain of thought : 마지막 output을 유도하기 위한 중간 단계를 말로 설명한 부분, 즉 추론 과정을 말로 하나씩 설명

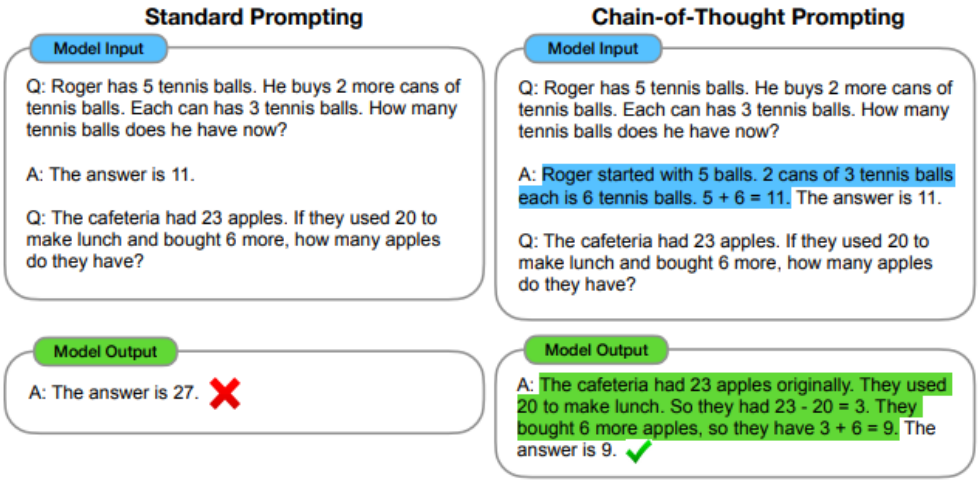


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

⇒ 답이 11이 나오게 되는 과정을 (chain-of-thought demonstrations) 제공함으로써 모델이 추론 과정을 이해하여 연산이 가능하게 됨.

---

2. Chain-of-Thought Prompting

일반적으로 수학 문제를 풀 때, 우리는 문제를 중간 단계로 분해하고 최종 답변을 내기 위해 각각의 문제를 해결해가는 방식을 사용한다.

이처럼, LLM에게 최종 답변을 도출하기 위한 중간 추론 단계를 프롬프트로 제공함으로써 추론 능력을 가지게 만든다.

chain-of-thought prompting은 여러 특성을 가지고 있다.

- chain-of-thought는 모델이 multi-step problem을 여러 intermediate steps으로 분해할 수 있도록 하며 이는 더 많은 추론 단계가 필요한 문제들에 추가적인 계산 과정을 할당할 수 있다.
- chain-of-thought는 모델이 어떻게 답을 도출하게 되었는지, 어느 부분이 잘못 되었는지를 디버깅 할 수 있게 만든다.
- 원칙적으로 이 방법은 언어를 통해 해결할 수 있는 모든 작업에 적용이 가능하다
- 충분히 큰 LLM에서는 간단하게 chain-of-thought sequence 예시를 포함하여 쉽게 chain-of-thought reasoning을 얻을 수 있다.

---

3. Arithmetic Reasoning

우리는 수학 문제를 푸는 것이 어렵지만 LLM은 종종 어려움을 겪는다. 해당 문제를 prompting을 통해 fine-tuning한 결과와 비슷한 성능을 확보했으며 어려운 벤치 마크 데이터에서는 최고 기록을 달성하기도 했다.

3.1 Experimental Setup

- 5가지 수학 문제 벤치 마크 데이터를 사용 : GSM8K / SVAMP / ASDiv / AQuA / MAWPS

Table 12: Summary of math word problem benchmarks we use in this paper with examples. N: number of evaluation examples.

Dataset	N	Example problem
GSM8K	1,319	Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?
SVAMP	1,000	Each pack of dvds costs 76 dollars. If there is a discount of 25 dollars on each pack. How much do you have to pay to buy each pack?
ASDiv	2,096	Ellen has six more balls than Marin. Marin has nine balls. How many balls does Ellen have?
AQuA	254	A car is being driven, in a straight line and at a uniform speed, towards the base of a vertical tower. The top of the tower is observed from the car and, in the process, it takes 10 minutes for the angle of elevation to change from 45° to 60°. After how much more time will this car reach the base of the tower? Answer Choices: (a) $5\sqrt{3}$ + 1 (b) $6\sqrt{3}$ + $\sqrt{2}$ (c) $7\sqrt{3}$ - 1 (d) $8\sqrt{3}$ - 2 (e) None of these
MAWPS: SingleOp	562	If there are 7 bottle caps in a box and Linda puts 7 more bottle caps inside, how many bottle caps are in the box?
MAWPS: SingleEq	508	Benny bought a soft drink for 2 dollars and 5 candy bars. He spent a total of 27 dollars. How much did each candy bar cost?
MAWPS: AddSub	395	There were 6 roses in the vase. Mary cut some roses from her flower garden. There are now 16 roses in the vase. How many roses did she cut?
MAWPS: MultiArith	600	The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with?

수학 문제에 대한 데이터셋들

- **Standard prompting** : baseline으로 standard few-shot prompting을 사용 (Language Models are Few-Shot Learners 논문에서 제시 된 방법)
  - input-output 예시를 in-context exemplars로 제공하는 방법
  - exemplars은 질문과 답변 형태로 제작

- **Chain-of-thought prompting**
  - figure 1에서 왼쪽처럼 모델이 질문에 대한 직접적인 답변을 제시하는 것과 다르게, 오른쪽처럼 사고 과정에 대한 예시를 제공
  - 대부분의 데이터셋이 evaluation split만 가지고 있으므로 총 8개의 few-shot exemplars을 직접 만들어 제공 (Table 20)
    - AQuA는 다중 선택 데이터이므로 4개의 예시와 solution을 사용

Table 20: Few-shot exemplars for full chain of thought prompt for math word problems. This set of exemplars was used for all math word problem datasets except AQuA.

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.

프롬프팅을 위한 8개의 chain-of-thought 예시

- prompt engineering을 거치지 않았으며 robustness에 대해서는 section 3.4 / 부록 A.2에서 추가로 다룰 예정
- GPT / LaMDA / PaLM / UL2 20B / Codex 모델에 대해서 실험을 진행
  - LaMDA의 경우, seed가 다르다고 해서 결과에 큰 변동성이 없었으므로 예시가 뒤섞인
  - 5개의 seed에 대한 평균 값을 사용 (큰 의미 있는 부분은 X)

3.2 Results

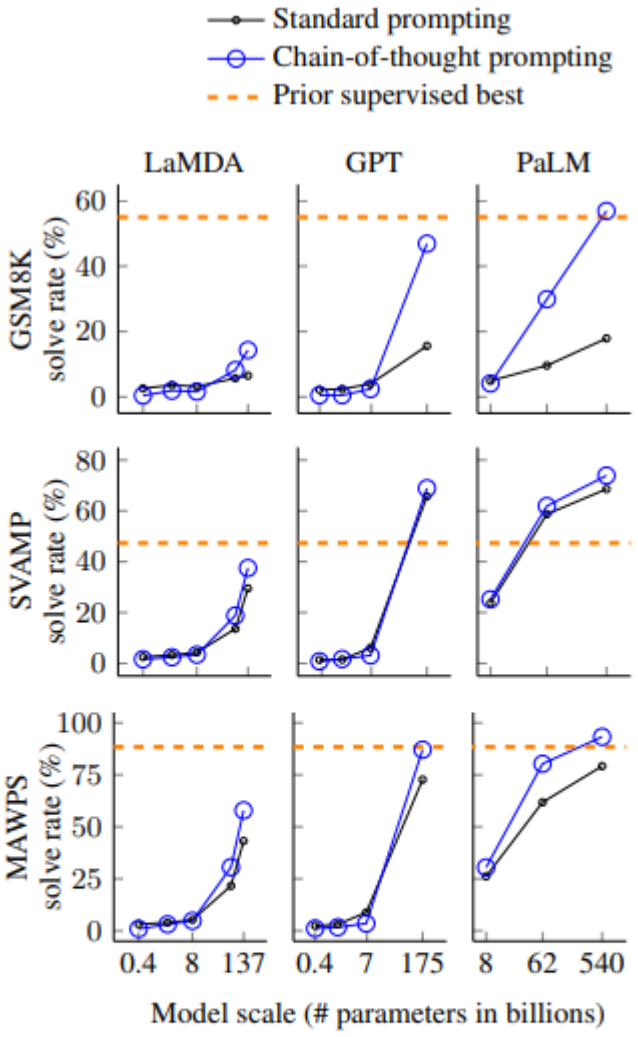


Figure 4: Chain-of-thought prompting enables large language models to solve challenging math problems. Notably, chain-of-thought reasoning is an emergent ability of increasing model scale. Prior best numbers are from Cobbe et al. (2021) for GSM8K, Jie et al. (2022) for SVAMP, and Lan et al. (2021) for MAWPS.

Figure 4는 가장 강력한 결과에 대한 요약을 의미한다. 크게 3가지의 중요한 결론이 있다.

1. Figure 4는 chain-of-thought prompting이 모델 규모가 증가함에 따라 나타나는 abililty이다.
    a. 즉, 작은 모델에는 긍정적인 영향을 기대하긴 어려우며, 100B parameter model과 함께 사용될 때만 성능이 향상된다.
    b. 본 연구에서 작은 모델이 유창하지만 비논리적인 chain-of-thought를 생성하여 표준 prompting보다 낮은 성능을 보이는 것을 확인
2. Chain-of-thought prompting은 문제가 더 복잡할수록 더 큰 성능 향상을 보인다.

a. GSM8K는 기본 성능이 가장 낮게 측정되는 데이터셋 → 가장 큰 모델에 대한 성능이 2배 이상 증가

b. 반면, 간단하게 해결할 수 있는 MAWPS의 SingleOp에서는 성능 향상이 없거나 오히려 떨어지는 경우가 발생

3. GPT, PaLM에 chain-of-thought를 적용하는 것은 fine-tuning을 사용한 기존 기술과 비교했을 때 훨씬 유리하다.

또한, CoT prompting이 효과적인지를 더 잘 이해하기 위해 모델이 생성하는 chain-of-thought를 수동으로 검토

- 정답을 맞춘 50개의 예시 중, 2개를 제외하고 모두 CoT로 접근
- CoT가 거의 정확했으나 사소한 실수 (계산 오류, 하나의 추론 단계가 생략) 등의 문제 발생
- 따라서, PaLM을 62B → 540B로 확장하면 해당 문제들을 일부 해결 가능

3.3 Ablation study

chain-of-thought 말고 다른 유형의 프롬프팅으로 동일한 성능 향상이 가능한가? → ablation study 시행

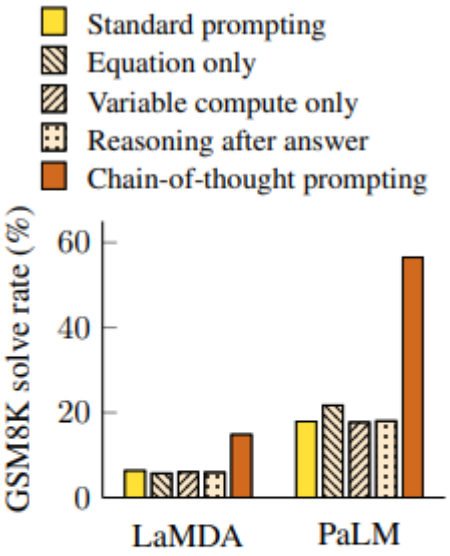


Figure 5: Ablation study for different variations of prompting using LaMDA 137B and PaLM 540B. Results for other datasets are given in Appendix Table 6 and Table 7.

- equation only : chain-of-thought는 답변을 생성하기 까지 여러 수학 방정식을 생성하기 때문에 수학 방정식만 출력하도록 프롬프팅 → 성능 향상 X

Table 6: Ablation and robustness results for arithmetic reasoning datasets. Chain of thought generally outperforms ablations by a large amount. "Equation only" performs in between standard prompting and chain of thought prompting, as it allows for intermediate reasoning steps via equations but does not leverage natural language. Chain of thought prompting has variance (as expected) when used with prompts written by different annotators or when using other exemplars, but still outperforms standard prompting by a large margin. Standard deviation shown is for different order of few-shot prompting exemplars, with five different random seeds. Results here are shown for LaMDA 137B, as additional queries for GPT-3 and PaLM are both limited and expensive.

	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

    - 하지만 one-step / two-step problem에서는 어느 정도 성능 향상이 있었음을 확인할 수 있다. (GSM8K를 제외한 나머지 데이터셋)
- Variable compute only : chain-of-thought가 모델이 더 어려운 문제에 더 많은 intermediate을 할당할 수 있게 한다는 점을 직관적으로 반영하여, 모델이 문제를 해결하기 위해 필요한 방정식의 문자 수와 동일한 dots (…)의 sequence를 제공
    - 성능 향상이 의미 없음, standard prompting과 유사한 결과
- Chain of thought after answer : prompting이 사전 훈련된 지식에 더 잘 접근하게 해준다는 점에서, 답변 후에 chain-of-thought를 제공하는 방식으로 테스트하여 실제 모델 답을 추론하는 데에 있어서 chain-of-thought가 필요한 지 실험
    - standard prompting과 유사한 결과

3.4 Robustness of Chain of Thought

prompt에 제공하는 예시에 대한 sensitivity는 주요 고려 사항이다. **Calibrate Before Use: Improving Few-Shot Performance of Language Models** 논문에 따르면 exemplars의 순서를 변화시키는 것만으로도 54% → 93.4%의 성능 차이를 보였다.

따라서, chain-of-thought를 연구자들을 제외하고도 다른 annotator가 작성한 prompt로 CoT의 robustness를 평가하고자 한다.

H Appendix: Alternate Annotators for MWP

Table 29: Few-shot exemplars for full chain of thought prompt for math word problems. These exemplars are the same as in Table 20, except that the chains of thought were written by a different annotator ("Annotator B" instead of "Annotator A"). Annotators were co-authors and familiar with the goal of chain of thought prompting.

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 21 trees now and there are 15 trees in the beginning, so the workers plant 21 - 15 = 6 trees. 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 3 cars in the beginning, 2 more arrive, so now there should be 3 + 2 = 5 cars. 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: Leah had 32 chocolates and her sister had 42, in total they have 32 + 42 = 74 chocolates. After they ate 35, now there are 74 - 35 = 39 chocolates. 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, but now he only has 12, so he gave Denny 20 - 12 = 8 lollipops. 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 got 2 toys each from his mom and dad, so he got 2 * 2 = 4 more, now he will have 5 + 4 = 9 toys. 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: 5 computers were installed from monday to thursday, so in total 5 * 4 = 20 computers are installed. 9 computers are there in the beginning, so now there are 20 + 9 = 29 computers. 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 and lost 23, so he has 58 - 23 = 35. After he lost 2 more, he has 35 - 2 = 33 balls now. The answer is 33.

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

A: 5 bagels for $3 each should cost 5 * 3 = 15 dollars. Olivia had $23 in the beginning, so now she has 23 - 15 = 8 dollars left. The answer is 8.

table 20에서 제시된 8개의 CoT에 대한 또 다른 prompt 예시

Training Verifiers to Solve Math Word Problems 논문에서 GSM8K에 대한 효율적인 prompt 제시 방법을 따라 새로운 예시를 작성 (intentionally concise style)

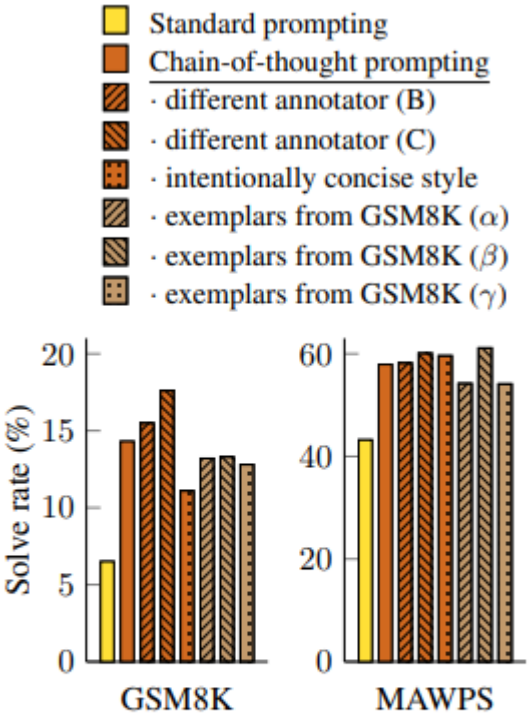


Figure 6: Chain-of-thought prompting has variance for different prompt examples (as expected) but outperforms standard prompting for various annotators as well as for different exemplars.

LaMDA → GSM8K, MAWPS에 대한 결과

Table 6: Ablation and robustness results for arithmetic reasoning datasets. Chain of thought generally outperforms ablations by a large amount. "Equation only" performs in between standard prompting and chain of thought prompting, as it allows for intermediate reasoning steps via equations but does not leverage natural language. Chain of thought prompting has variance (as expected) when used with prompts written by different annotators or when using other exemplars, but still outperforms standard prompting by a large margin. Standard deviation shown is for different order of few-shot prompting exemplars, with five different random seeds. Results here are shown for LaMDA 137B, as additional queries for GPT-3 and PaLM are both limited and expensive.

	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

Table 7: Ablation and robustness results for four datasets in commonsense and symbolic reasoning. Chain of thought generally outperforms ablations by a large amount. Chain of thought prompting has variance (as expected) when used with prompts written by different annotators or when using other exemplars, but still outperforms standard prompting by a large margin. Standard deviation shown is for different order of few-shot prompting exemplars, with five different random seeds. Results here are shown for LaMDA 137B, as additional queries for GPT-3 and PaLM are both limited and expensive. The exception is that we run SayCan using PaLM here, as the SayCan evaluation set is only 120 examples and therefore less expensive to run multiple times.

	Commonsense			Symbolic	
	Date	Sports	SayCan	Concat	Coin
Standard prompting	21.5 ±0.6	59.5 ±3.0	80.8 ±1.8	5.8 ±0.6	49.0 ±2.1
Chain of thought prompting	26.8 ±2.1	85.8 ±1.8	91.7 ±1.4	77.5 ±3.8	99.6 ±0.3
Ablations					
· variable compute only	21.3 ±0.7	61.6 ±2.2	74.2 ±2.3	7.2 ±1.6	50.7 ±0.7
· reasoning after answer	20.9 ±1.0	63.0 ±2.0	83.3 ±0.6	0.0 ±0.0	50.2 ±0.5
Robustness					
· different annotator (B)	27.4 ±1.7	75.4 ±2.7	88.3 ±1.4	76.0 ±1.9	77.5 ±7.9
· different annotator (C)	25.5 ±2.5	81.1 ±3.6	85.0 ±1.8	68.1 ±2.2	71.4 ±11.1

다양한 데이터셋에 대한 결과

다양한 annotation에 따라 조금의 성능 차이는 보이지만 모든 CoT는 standard prompting 보다 훨씬 좋은 성능을 이끌어 낸다. 또한 이 결과는 특정 언어 스타일에 의존하지 않고 CoT를 사용할 수 있음을 의미한다.

GSM8K train dataset에서 무작위로 추출한 예시 3개로 실험 진행 → CoT와 비슷한 성능을 보이면서 CoT의 성능 입증

▼ Appendix A.2 : What is the role of prompt engineering?

- prompting에 대한 key consideration은 prompt에 대한 sensitivity이다.
    - 언어 모델에 대한 prompt가 예상치 못한 방식으로 영향을 미친다는 논문, 연구들이 많다. (예시로, **Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?** 논문에서 ground truth demo가 실제로 필요하지 않다는 연구로 인해 prompt 작성 예시에 대한 변화를 다룸)
- CoT를 만드는 일반적인 방법은 train dataset에서 8개의 예시를 가져와 최종 답변으로 이어지는 추론 과정을 여러 단계로 분해하는 것

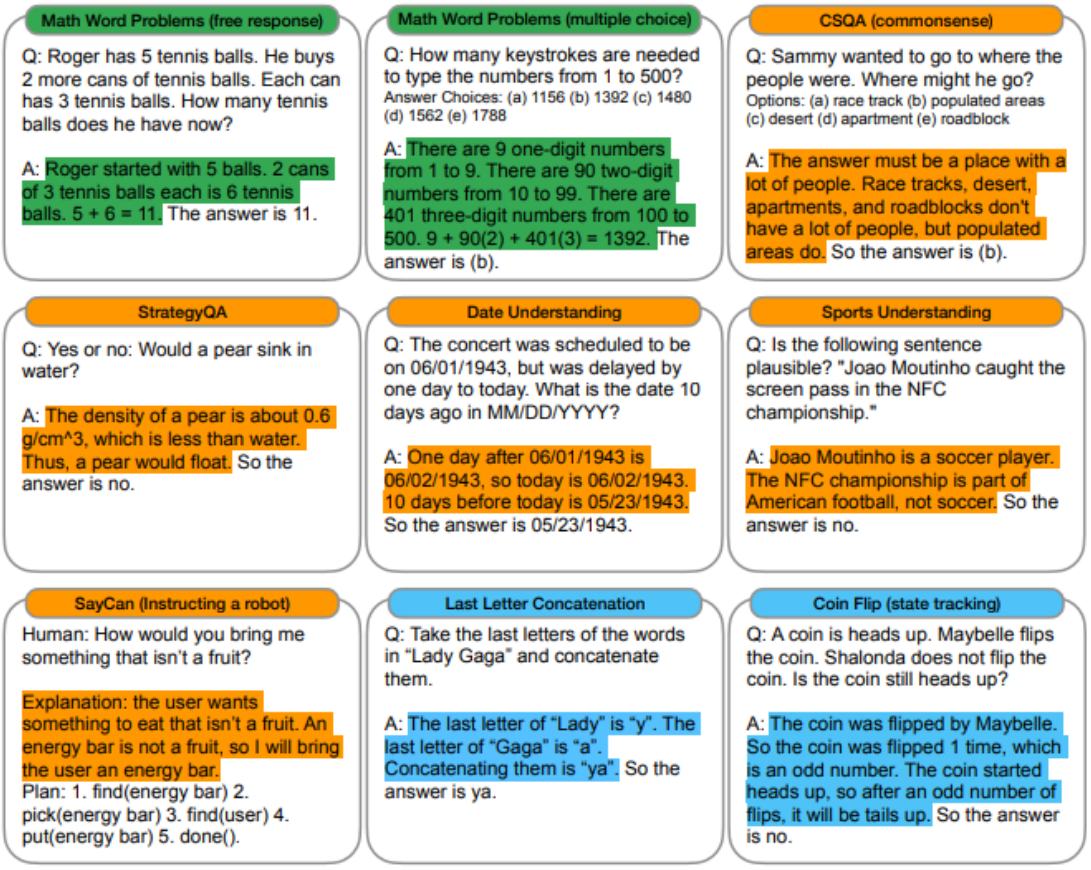


Figure 3: Examples of ⟨input, chain of thought, output⟩ triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

프롬프트 작성 예시 및 전체 작성 예시는 appendix G 참

- Different annotators : 여러 사람이 작성한 prompt로 실험 진행했을 때도 충분히 좋은 성능을 얻을 수 있었다.
- Annotators without machine learning background : ML에 대한 지식이 없는 사람들이 작성한 prompt에 대해서도 CoT가 잘 작동했다.

- Different exemplars : Table 6에서 실험한 예시 외에도 다른 예시에서도 좋은 성능을 보여줄 수 있었다.
- Different order of exemplars : 이전 연구에 따르면 prompt의 순서조차 모델 성능에 영향, table 6,7에서 여러 예시들로부터 성능의 표준 편차를 확인할 수 있고 모든 경우에서 상대적으로 표준 편차가 미비한 것으로 보아 순서에 따른 영향이 작은 것으로 보임.
    - 예외도 있었는데, 동전 던지기와 같은 부분에서는 예시의 순서에 대해 민감성이 높았는데 그 이유로, 이전 다른 연구에서 “분류 작업에서 같은 카테고리의 많은 예시가 연속되면 모델 출력에 bias가 생길 수 있다” 라는 것으로 분석.
- Different number of exemplars : 예시의 수가 다양하더라도 CoT를 통해 얻는 benefit이 일반적으로 유지되는 것을 확인할 수 있었다.
- Different language models : 특정 모델에서 잘 작동하는 prompt가 다른 LLM에서도 잘 작동하는 지에 대한 연구 → LaMDA, GPT-3, PaLM에서 모두 성능을 향상
    - 하지만 CoT를 통해 얻는 이득이 모델 간에 완벽하게 전달되지 않는 것 (호환되지 않는다는 뜻)은 한계점
    - 따라서 각 훈련 데이터와 모델 아키텍처가 CoT로부터 어떻게 성능 향상에 대한 영향을 미치는 지에 대한 추가 연구가 필요
- Prompt engineering still matters, though : arithmetic 경우에 CoT가 좋은 성능을 보일지라도 prompt engineering을 통해 성능을 크게 향상시킬 수 있다.
    - 일반적으로 CoT가 standard prompting보다 성능이 뛰어나지만 high variance가 존재
    - 또한, prompt engineering이 필수적인 작업도 존재
        - 3명 중, 2명은 모델이 순서를 바꿔서 대답하는 것을 하지 못했지만 1명은 해당 작업을 수행하는 CoT를 작성할 수 있었다.
        - **따라서 robust한 prompt를 작성하는 것을 future work로 둘 수 있다.**

---

4. Commonsense Reasoning

Chain-of-thought (CoT)는 arithmetic problem에 특히 적합하지만, CoT의 linguistic property는 commonsense reasoning과 같은 문제에도 광범위하게 적용될 수 있다.

- Benchmark dataset 5개 사용
    - CSQA : 세계에 대한 상식적인 질문 + 사전 지식이 필요한 복잡한 의미론

- strategyQA : multi-step strategy를 추론
- BIG-bench
  - Date Understanding
  - Sports Understanding
- SayCan

- **Prompts :** 이전 section들과 동일하게 예시를 무작위로 선택하고 해당 CoT를 manually로 구성하여 사용

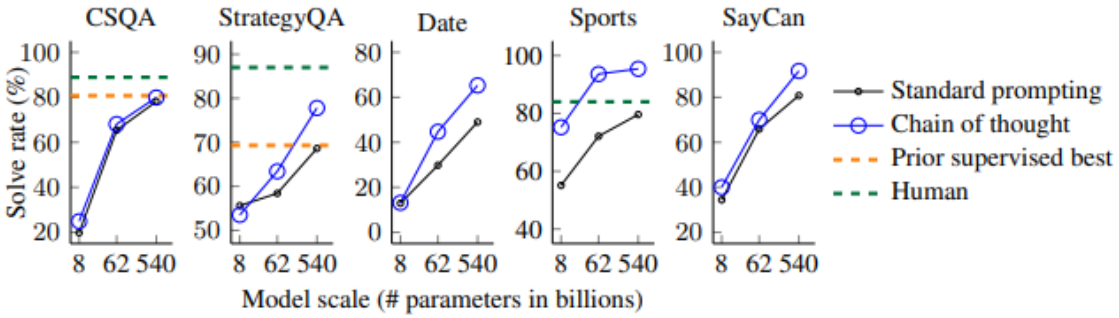


Figure 7: Chain-of-thought prompting also improves the commonsense reasoning abilities of language models. The language model shown here is PaLM. Prior best numbers are from the leaderboards of CSQA (Talmor et al., 2019) and StrategyQA (Geva et al., 2021) (single-model only, as of May 5, 2022). Additional results using various sizes of LaMDA, GPT-3, and PaLM are shown in Table 4.

- **Results :** PaLM에 대한 결과 (figure 7)
  - 기존 fine-tuning과 비슷한 결과부터 시작해서 월등히 뛰어난 성능, 심지어 인간을 능가하는 성능까지 보여줌
  - 이를 통해 CoT prompting이 commonsense reasoning을 요구하는 작업에서도 충분히 좋은 성능을 보여줄 수 있음을 의미

---

5. Symbolic Reasoning

사람은 풀기 쉽지만, 언어 모델은 풀기 어려운 task

- 각 단어의 끝 문자를 모으는 것 (ex. Amy Brown → ‘yn’)
- 동전의 초기 상태를 주고, 몇 번의 행동 이후에 동전의 상태를 유추

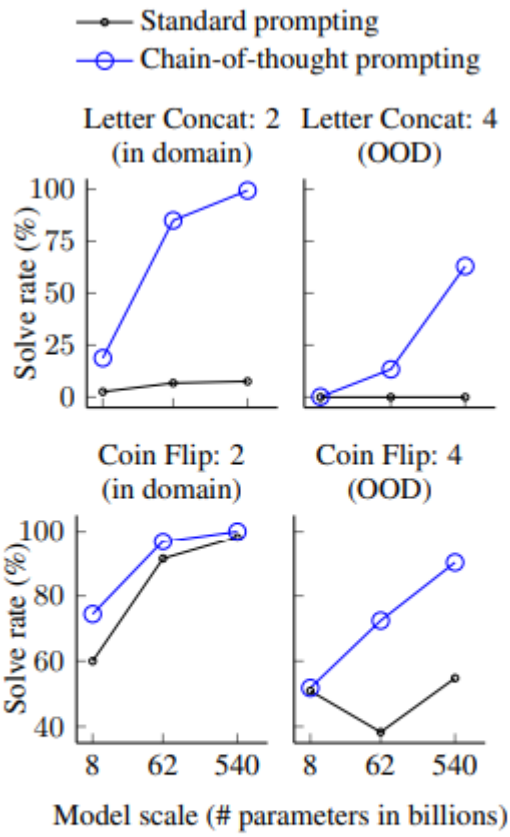


Figure 8: Using chain-of-thought prompting facilitates generalization to longer sequences in two symbolic reasoning tasks.

작은 모델들에서는 여전히 실패하는 모습 → 충분한 규모의 언어 모델에 대해서는 CoT를 넘어서서 length generalization도 가능.

---

6. Discussion

LLM에서 multi-step reasoning을 유도하기 위한 간단한 메커니즘을 제시

- arithmetic reasoning에서 성능을 크게 향상시킴
  - 기존 연구보다 훨씬 강하며 robust하다.
- commonsense reasoning 실험을 통해 일반적으로 적용 가능함을 제시
- symbolic reasoning에 대해 CoT가 더 긴 sequence length에 대해 OOD generalization 가능
  - OOD generalization to longer sequence length?

    - OOD generalization : Out Of Domain (OOD), 즉 훈련 중에 본 적이 없는 데이터 또는 상황에 대해서 얼마나 잘 수행할 수 있는지
    - longer sequence length : 모델이 처리해야 할 데이터의 길이나 복잡성이 증가

⇒ CoT를 단순하게 기존 LLM에 prompting하여 쉽게 성능 향상 가능

CoT는 LLM이 성공적으로 수행할 수 있는 task를 확장 → standard prompting은 LLM의 하한선만 제공, CoT와 기타 method들을 통해 얼마든지 성능을 향상 시킬 수 있다는 점

연구를 진행하면서 궁금한 점

- 모델의 규모가 증가함에 따라 얼마나 더 성능이 향상될지
- 다른 prompting 방법으로 얼마나 task 확장이 가능한지

Limitations

- CoT가 사람의 사고 과정을 모방한다고 해도 neural network가 실제로 “reasoning”을 하는 것인지에 대한 여부는 모호함
- few-shot에 대한 augmentation을 위해서는 cost가 많이 들진 않지만 fine-tuning을 위해 CoT를 작성하는 것은 cost가 높다.
- 올바른 reasoning path가 보장되지 않아서 정답과 오답을 모두 이끌어낼 수 있다.
    - 따라서 언어 모델의 factual generation을 향상시키는 것을 future work
- 큰 모델에 대해서만 잘 작동 → 작은 모델에서도 reasoning을 유도하는 방법을 탐구하는 추가 연구가 필요

---

7. Related Work

Appendix C 참고

▼ Appendix C

C.1 prompting

- LLM에 prompting을 통해 수행 능력을 향상 시키는 작업에 대한 관심 증가
    - input prompt를 최적화하여 성능 향상

- 최근 연구는 task에 대한 설명을 제공하면서 언어 모델이 작업을 수행하는 능력을 향상
    - input-output 데이터에 meta data 추가
- 또 다른 연구 방향으로는 언어 모델이 생성한 output을 순차적으로 결합
    - human-computer interaction (HCI) work

C.2 Natural language explanations

- 모델 해석 가능성을 향상 시키기 위해 Natural Language Explanations (NLE) 사용
    - final prediction 이 후, explanations을 생성 ↔ CoT는 최종 답변을 내기 전, explanations을 생성

C.3 Program synthesis and execution

- intermediate reasoning step은 program synthesis / execution에 많이 사용됨
- 특정 task에 대한 domain-specific한 요소들을 text-to-text NLP task로 일반화하는 것이 해당 논문의 contribution

C.4 Numeric and logical reasoning

- numeric reasoning abilities를 위해서 BERT에 executable operations의 집합을 넣거나 graph neural network를 포함 시키는 등의 작업 수행

C.5 Intermediate language steps

- 기존에는 intermediate step을 생성하는 능력을 training or fine-tuning을 통해 얻을 수 있었음
- 본 연구는 충분한 크기의 언어 모델에 prompting을 통해 해당 능력을 얻음
    - prompting setup이 굉장히 중요한데, 이는 많은 수의 labeled annotations 없이 intermediate step reasoning이 가능하도록 하며 단일 모델이 어떤 gradient update없이 여러 reasoning task를 수행하기 때문이다.

---

8. Conclusions

- LLM의 reasoning ability를 향상시키기 위해 간단하면서도 광범위하게 적용할 수 있는 chain-of-thought prompting 제시
- 여러 문제에 대한 실험을 통해 chain-of-thought reasoning이 모델 규모가 커짐에 따라 생기는 특성임을 확인
- LLM이 수행할 수 있는 reasoning task의 범위를 넓혀 language-based approach에 대한 future work에 많은 영감을 줄 것