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

Chain-of-Thought(CoT, 사고연쇄) 프롬프팅이 LLM에 추론을 유도한다

Abstract

- Chain-of-Thought(CoT) 프롬프팅이 LLM의 추론 능력을 어떻게 향상시키는지 알아본다

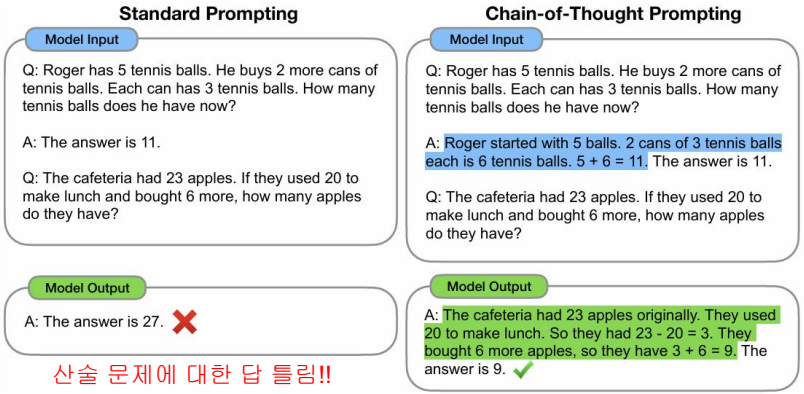
- 실험을 통해 CoT 프롬프팅이 세 문제(산술, 상식, 기호 추론 문제)에 대해 성능 향상시킨다는 것 보여줌

chain-of-thought : LLM이 복잡한 문제를 해결할 때,
바로 최종 답을 내놓지 않고 단계별로 추론하는 과정을 거쳐 답을 도출하도록 유도하는 프롬프팅 기술

Standard Prompting : 질문 - 답

Chain-of-Thought Prompting : 질문 - 사고과정 - 답

표준 프롬프팅 vs CoT 프롬프팅



1. Introduction

- 언어모델의 크기 확장하면 → 장점 : 성능 향상, 샘플 효율성 증가

  → 단점 : 산술, 상식, 기호추론 문제같은 어려운 문제 해결의 성능을 향상시키기에는

  충분하지 않다!

- LLM의 추론 능력 극대화하는 법

  방법1. generate natural language rationales : 정답에 도달하는 근거를

  자연어로 설명

  방법2. in-context few-shot : [질문-정답] 예시를 프롬프트로 제시

- 하지만 이 두 방법에는 한계가 존재함

  방법1. 근거 만들어야 됨 → finetuning(학습)시키려니 데이터 만드는 비용 비쌈!!

  방법2. 기존처럼 [문제-답] 예시만 보여주면 → 복잡한 추론문제 해결 못함(과정을 모르니까)

- 만약 두 방법을 합친다면…? 문제 해결!!!
- 본 논문에서는, 언어모델이 추론 문제에 대해 few-shot 프롬프팅을 하는 능력 탐구

  input, chain-of-thought(사고과정), output으로 구성된 프롬프트 제시하기

  (문제 - 풀이과정 - 답)

few-shot 프롬프팅 : AI 언어 모델(LLM)에게 질문이나 명령을 내릴 때, 원하는 출력 형식이나 답변의 예시(shot)를 1~5개 정도(few) 제공하여 모델의 정확도와 성능을 향상시키는 프롬프트 엔지니어링 기술

2. Chain-of-Thought(CoT) Prompting

ex) 내가 빵 10개 중에 2개를 먹으면 8개가 남고, 또 5개를 먹으면 3개가 남고…

- 언어모델에게 이러한 사고과정 (= 일련의 추론 단계를 생성하는 능력)을 부여하는 것이 목표

that it would have otherwise gotten incorrect. The chain of thought in this case resembles a solution and can interpreted as one, but we still opt to call it a chain of thought to better capture the idea that it mimics a step-by-step thought process for arriving at the answer (and also, solutions/explanations typically come *after* the final answer (Narang et al., 2020; Wiegreffe et al., 2022; Lampinen et al., 2022, *inter alia*)).

- chain-of-thought(생각의 사슬)는 풀이과정 해설지가 아님

  정답이 뭔지 모르는 상태에서 정답에 도달하기 위해 단계별로 사고하는 과정을 따라함

- CoT 프롬프팅의 특징

1. 복잡한 문제를 여러 중간 과정으로 나눠서 풀기 가능

   → 어려운 문제 쪼개서 풀기

2. 모델이 정답에 도달한 과정을 보여주고, 그 과정이 틀린 부분을 찾아 디버깅할 수 있음

   → 틀린 이유 찾기

3. 사람이 언어로 풀 수 있는 모든 문제를 CoT로 풀 수 있을 정도로 범용성 높음

   → 어디든 적용 가능

4. 이미 만들어진 모델에 CoT 과정 예시만 넣어주면 바로 사용 가능

   → 쓰기 쉬움

3. Arithmetic Reasoning 산술 추론에 대한 CoT의 유용성

- 3.1 실험설계

Benchmarks : 모델 평가하기 위해 GSM8K, SVAMP, ASDiv, AQuA, MAWPS 5개의 수학문제 모음집 사용

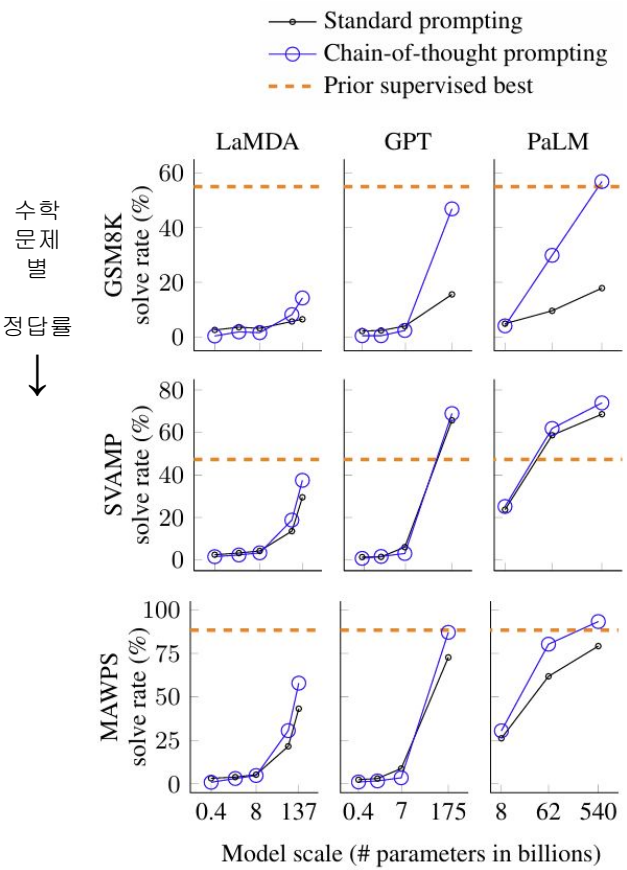
테스트한 AI 모델 : GPT-3, LaMDA, PaLM, UL2, Codex

Standard Prompting (대조군) : 프롬프트에 [질문-정답] 예시 넣음

CoT Prompting (실험군) : 프롬프트에 [질문-풀이과정-정답] 예시 넣음

⇒ AI 모델들에게 수학문제를 풀게 시키고 뭐가 더 나은지 비교하기

- 3.2 결과



1. CoT 프롬프팅은 100B(1000억)개 이상의 파라미터를 가진 모델에 성능향상을 가져옴

2. CoT는 복잡한 문제에 더 큰 성능향상 가져옴

GSM8K(복잡한 문제) : Sp보다 CoT 성능 2배 가까이 좋음
MAWPS(쉬운 문제) : Sp나 CoT나 성능 비슷함

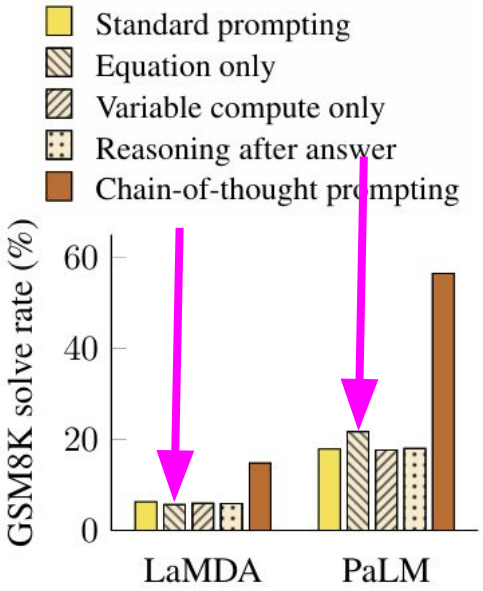
3. PaLM 540B 모델에 CoT 프롬프팅을 적용했을 때 기존의 최고성능 지도학습 모델을 상회하는 결과

모델이 생성한 chain-of-thougnt를 사람이 검사했을 때, 무작위로 선정한 50개의 답변 모두 논리적, 수학적으로 정확함 (단 2개만 우연)

결론 ⇒ 충분한 규모를 가진 LLM에 CoT 프롬프팅을 적용하면, 별도의 추가학습 없이 복잡한 문제를 해결하는 능력이 상승한다. (특화된 전용모델 수준으로)

- 3.3 Ablation Study : 머신러닝/딥러닝 모델에서 특정 구성 요소를 하나씩 제거하거나 변경하며 모델 성능을 비교 분석하는 실험

다른 유형의 프롬프팅을 통해서도 같은 성능향상을 얻을 수 있을까?



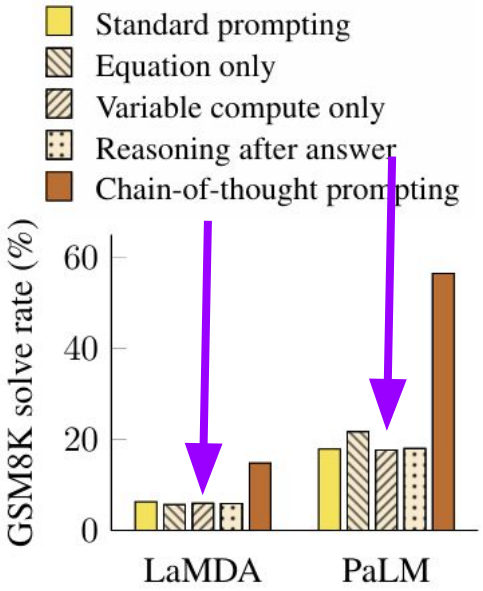
- 중간과정 다 필요없고, 공식만 잘 세우면 성능 좋은 거 아니야?

사과 5개 사고, 한 박스에 3개가 들어있는 걸 2개 더 샀다.
이 중간 논리 없이,
5 + (3 x 2) = 11가 아닌 **5 + 6 = 11**에 도달해야 함

결과 : 기존 방식(Standard Prompting)과 별 차이 없음

⇒ 복잡한 문제에 대해 자연어 사고과정 없이, 공식을 바로 도출해내기 어렵다. 즉, 5+6=11에 도달하지 못한다.

- 3.3 Ablation Study : 머신러닝/딥러닝 모델에서 특정 구성 요소를 하나씩 제거하거나 변경하며 모델 성능을 비교 분석하는 실험



- 계산시간 많이 줘서 그런거 아니야?

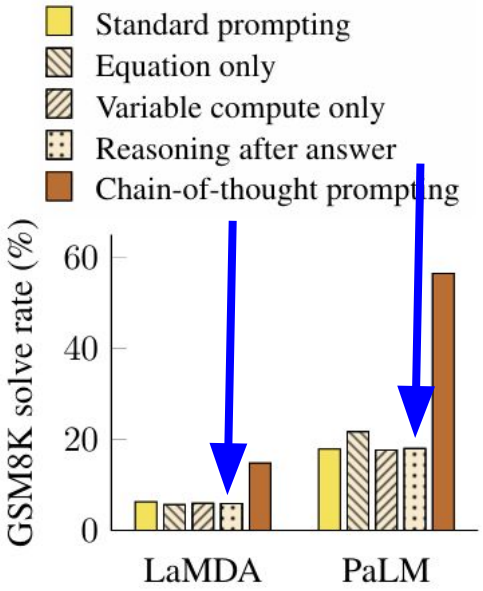
CoT는 정답 전에 긴 풀이과정 생성 = 계산량 많음
자연어 풀이과정 자체가 중요할까? 아니면
토큰(글자)를 많이 생성하면서 계산시간을 버는것이 중요할까?

풀이과정 글자 수 만큼 … 찍기 (질문 → …. → 정답) = 계산시간 줄여보기

결과 : 차이 거의 없음

⇒ 계산 시간 중요하지 X
자연어를 통한 단계별 사고과정이 중요 O

- 3.3 Ablation Study : 머신러닝/딥러닝 모델에서 특정 구성 요소를 하나씩 제거하거나 변경하며 모델 성능을 비교 분석하는 실험



- 단순히 사전학습된 지식을 사용하는거 아니야?

모델이 논리적으로 사고하는게 아니라, 학습된 지식을 사용하는 것이 아닌가?

질문 → 답 → **풀이과정** 생성하도록 프롬프트 구성

결과 : 기존 프롬프팅보다 성능 비슷하거나 낮음

⇒ 정답 생성 전에 추론과정을 거쳐야 성능이 높아진다

- 3.4 Robustness of Chain of Thought

  CoT가 특정 조건에서만 유용한게 아니고, 안정적이고 강력한가?

  프롬프트 작성자가 달라도
  제시된 [질문 - 중간 풀이 과정 - 정답] 예시가 달라도
  그 예시의 순서와 개수가 달라도

  CoT는 성능향상을 일으킴

4. Commonsense Reasoning

상식 추론에 대한 CoT의 유용성

CoT는 언어 기반이기 때문에, 상식 문제에도 적용이 가능하다.

- Benchmarks

  상식추론 문제 5가지 데이터셋 사용 : CSQA, StrategyQA, Date, Sports, SayCan

- Prompts

  실험 방법 : 마찬가지로 Few-shot 방식 사용(CoT 예시 제시하는 거)

CSQA : 세상에 대한 상식적 질문, 사전지식 필요함
StrategyQA : 질문에 답하기 위해 여러 단계의 전략 세워야 하는 문제
Date : 문맥 속에서 날짜를 추론하는 문제
Sports : 스포츠와 관련된 문장이 상식적으로 말이되는지 판단하는 문제
SayCan : 사람이 내린 명령을 로봇이 수행할 수 있는 행동 순서로 매핑하는 문제

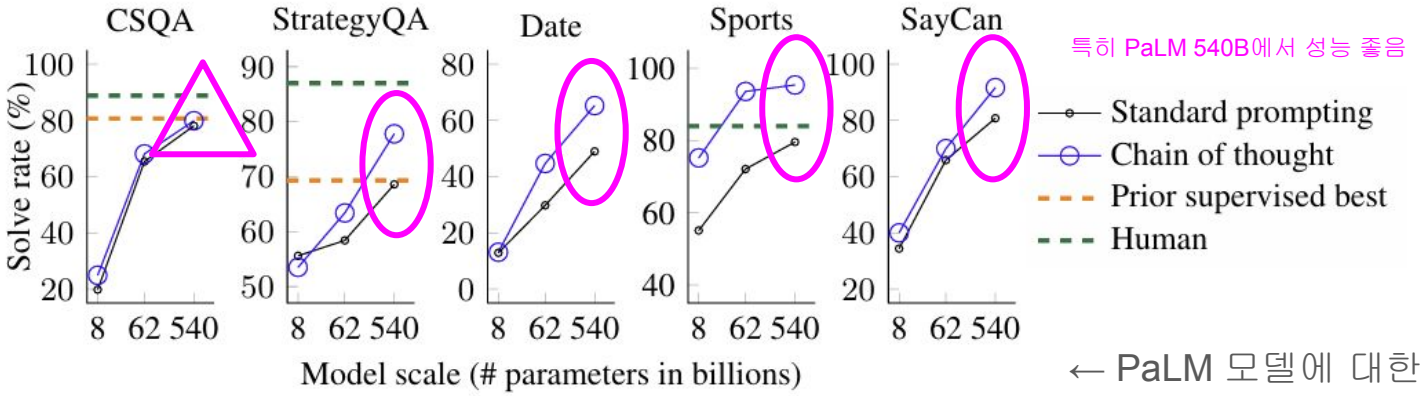
- Results

모델의 크기가 커질수록 Standard Prompting도 성능 좋아짐

하지만, CoT 썼을 때, 성능 향상의 폭이 훨씬 큼

다만, CSQA에서는 성능 향상 미미 (왜냐하면, CSQA는 그냥 배경지식 질문이지 추론을 요구하진 않아서)

⇒ CoT가 만능은 아님!! 추론 단계가 복잡할 때 유용함



특히 PaLM 540B에서 성능 좋음

← PaLM 모델에 대한 결과

5. Symbolic Reasoning 기호 추론에 대한 CoT의 유용성

기호 추론 : AI가 데이터를 처리할 때 숫자나 통계가 아닌, 인간이 이해할 수 있는 기호, 개념, 논리적 규칙을 사용하여 사고하는 방식

Standard Prompting : AI는 few-shot(예시)에서 보여준 것보다 더 길고 복잡한 문제가 나오면 틀리는 경우가 많음

CoT Prompting : 스스로 사과과정을 확장해 예시보다 복잡한 단계의 문제 해결함

⇒ CoT는 문제를 푸는 방식을 가르쳐준 것

- Tasks

문제1. 이름 끝 글자 합치기

ex) Amy Brown→ yn

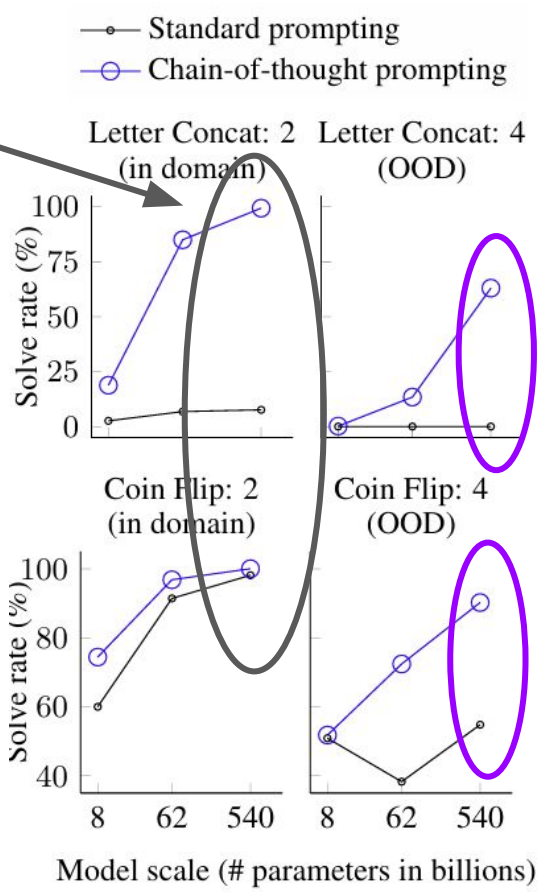
단어 2개인 이름 합치는 방법만 예시로 제시하고, 문제로는 단어 4개인 이름 합쳐라

문제2. 동전 던지기

ex) 동전이 앞면이다. P가 뒤집고, O는 안 뒤집음 → 동전은 뒷면임

동전 2번만 던지는 예시 제시하고, 문제로는 4번 던지고 앞인지 뒤인지 맞혀라

2단계 문제는
이미 풀 수 있음



예시로 제시한 것보다 많은 단계 수를 가진 문제에 대해서
CoT 프롬프팅을 사용한 모델의 성능이 훨씬 좋음

- Results

⇒ CoT의 Length Generalization(길이 일반화) 능력

2단계 풀이과정을 보고, 4단계, 여러 단계의 문제를 풀어낼 수 있음 = 응용력 높음

하지만 여전히 100B(파라미터가 1000억개) 이하의 모델에서는 성능이 좋지 않음

6. Discussion

- 성과

앞선 모든 그래프에서 확인했듯이 Standard Prompting은 그저 성능의 하한선일 뿐

CoT Prompting를 통해 성능 향상 가능

- 한계

1. 모델이 인간처럼 진짜 '생각'을 하는건지, 아님 단순히 흉내를 내는건지?

2. few-shot은 비용 적음, 근데 fine-tuning(대규모 학습)을 위해 [문제-풀이과정-정답] 만드는 것은 비용이 많이 듦

3. 추론과정이 항상 맞는건 아님

4. CoT는 LLM에서만 나타나는 특성 (작은 모델에서도 가능하게 하는 연구 필요)

8. Conclusions

- LLM에 CoT 프롬프팅을 적용

  기존의 한계(Standard Prompting의 성능)를 뛰어넘는 추론능력