

Self-Consistency Improves Chain of Thought Reasoning In Language Models

목차 · 발표 흐름

논문의 기여에서 한계와 후속 연구까지, 여섯 단계로 전개한다.

01 주제

한 문장 기여와 대표 성능 개선

02 문제 및 동기

Greedy decoding의 구조적 한계

03 핵심 메커니즘

Sample and Marginalize의 구조

04 핵심 결과

경로 다양성이 만드는 성능 이득

05 한계점

체계적 오류와 다수결의 취약성

06 추가 연구

후속 연구 · test-time scaling · 에이전트

1. 주제

Self-Consistency Improves Chain of Thought Reasoning in Language Models · Wang et al. · ICLR 2023

arXiv:2203.11171v4 [cs.CL] 7 Mar 2023

Published as a conference paper at ICLR 2023

SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS

Xuezhi Wang^{†‡} Jason Wei[†] Dale Schuurmans[†] Quoc Le[‡] Ed H. Chi[†]
Sharan Narang[†] Aakanksha Chowdhery[†] Denny Zhou^{†§}

[†]Google Research, Brain Team

[‡]xuezhiw@google.com, [§]dennyzhou@google.com

ABSTRACT

Chain-of-thought prompting combined with pre-trained large language models has achieved encouraging results on complex reasoning tasks. In this paper, we propose a new decoding strategy, *self-consistency*, to replace the naive greedy decoding used in chain-of-thought prompting. It first samples a diverse set of reasoning paths instead of only taking the greedy one, and then selects the most consistent answer by marginalizing out the sampled reasoning paths. Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer. Our extensive empirical evaluation shows that self-consistency boosts the performance of chain-of-thought prompting with a striking margin on a range of popular arithmetic and commonsense reasoning benchmarks, including GSM8K (+17.9%), SVAMP (+11.0%), AQuA (+12.2%), StrategyQA (+6.4%) and ARC-challenge (+3.9%).

1 INTRODUCTION

Although language models have demonstrated remarkable success across a range of NLP tasks, their ability to demonstrate reasoning is often seen as a limitation, which cannot be overcome solely by increasing model scale (Rae et al., 2021; BIG-bench collaboration, 2021, *inter alia*). In an effort to address this shortcoming, Wei et al. (2022) have proposed *chain-of-thought prompting*, where a language model is prompted to generate a series of short sentences that mimic the reasoning process a person might employ in solving a task. For example, given the question “If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?”, instead of directly responding with “5”, a language model would be prompted to respond with the entire chain-of-thought: “There are 3 cars in the parking lot already. 2 more arrive. Now there are 3 + 2 = 5 cars. The answer is 5.”. It has been observed that chain-of-thought prompting significantly improves model performance across a variety of multi-step reasoning tasks (Wei et al., 2022).

In this paper, we introduce a novel decoding strategy called *self-consistency* to replace the greedy decoding strategy used in chain-of-thought prompting (Wei et al., 2022), that further improves language models’ reasoning performance by a significant margin. Self-consistency leverages the intuition that complex reasoning tasks typically admit multiple reasoning paths that reach a correct answer (Stanovich & West, 2000). The more that deliberate thinking and analysis is required for a problem (Evans, 2010), the greater the diversity of reasoning paths that can recover the answer.

Figure 1 illustrates the self-consistency method with an example. We first prompt the language model with chain-of-thought prompting, then instead of greedily decoding the optimal reasoning path, we propose a “sample-and-marginalize” decoding procedure: we first *sample* from the language model’s decoder to generate a *diverse* set of reasoning paths; each reasoning path might lead to a different final answer, so we determine the optimal answer by *marginalizing out* the sampled reasoning paths to find the most consistent answer in the final answer set. Such an approach is analogous to the human experience that if multiple different ways of thinking lead to the same answer, one has greater confidence that the final answer is correct. Compared to other decoding methods, self-consistency avoids the repetitiveness and local-optimality that plague greedy decoding, while mitigating the stochasticity of a single sampled generation.

CORE CONTRIBUTION

CoT 추론에서 기존의 Greedy 디코딩을
여러 추론경로를 샘플링해 다수결하는
Self-Consistency로 대체하여
추론 정확도를 높인다.

PaLM-540B 기준 · 정확도 향상

GSM8K 56.5 → **74.4** (+17.9) · **AQuA** 35.8 → **48.3**
(+12.5)

SVAMP 79.0 → **86.6** (+7.6)

모델 가중치 · 프롬프트 · 파인튜닝은 그대로 두고, 토큰 선택 규칙만 바꾼다.

2. 문제 및 동기

Greedy decoding의 구조적 결함

Greedy 디코딩 : 매번 확률이 가장 높은 토큰을 선택

-> 오류의 반복

-> 지역 최적에 빠지기 쉽다

-> 별도의 검증 비용 발생(검증기, 재순위 지정기)

[지역 최적 예시]

문장 전체 확률 : 각 토큰 확률의 곱

A. $0.6 * 0.3 = 0.18$

B. $0.4 * 0.9 = 0.36$

B가 더 좋은 문장이지만 Greedy는 첫 순간에 좋은 A를 고른다.

논문의 직관

"복잡한 문제는 해결 방법이 다양하다"

“올바른 추론들은 최종 답이 일치하고, 틀린 추론들은 흩어진다”

-> 여러 추론경로를 만들어 답을 모아보고, 가장 일관된 답을 고르자!

Self-Consistency를 통한 해결

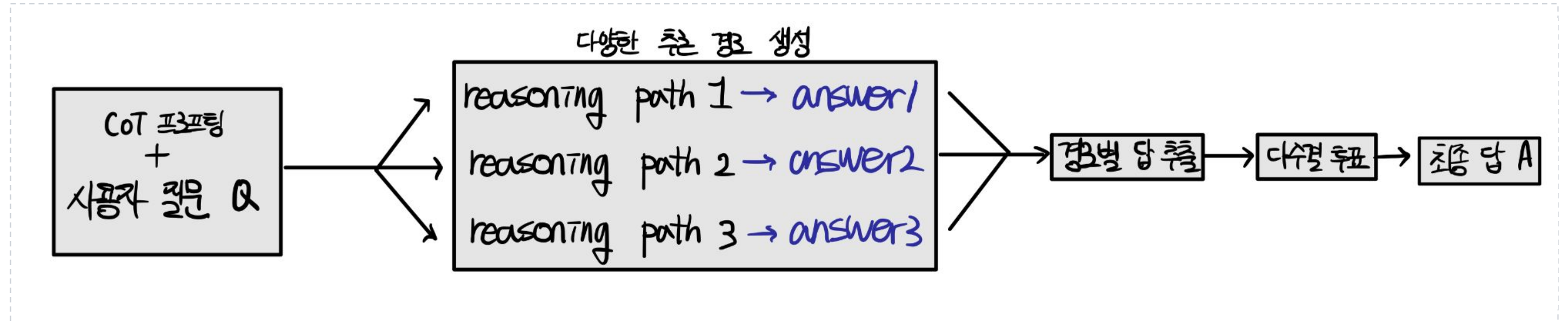
Self-Consistency : 여러 추론경로를 생성 후, 가장 일관된 답을 선택

- 매번 가장 높은 토큰이 아니라, 확률적으로 토큰을 선택
 - like(23%), love(13%), do(18%)

-> 샘플링을 통해 오류의 반복, 지역 최적 문제 해결

-> 정답 레이블/추가 학습/파인튜닝 불필요

3. 핵심 메커니즘



알고리즘 진행

1. CoT프롬프팅과 함께 사용자의 질문 input
2. 질문에 대한 다양한 풀이 과정 n개 샘플링
3. 각 경로마다 생성된 답 추출
4. 다수결 투표로 가장 일관된 답 output

3. 핵심 메커니즘

Greedy 디코딩

Q. 주차장에 차가 3대 있다. 5분 뒤 3대의 차가
출고 있다. 5분 뒤 주차장에 차가 몇대?

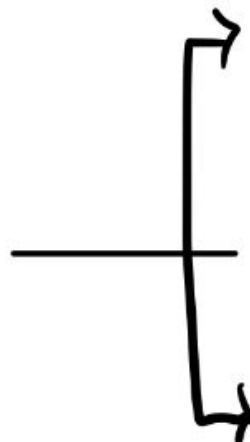


1. 주차장에 3대, 5분 뒤 3대 들어옴,
· $3+5+3=11$

$a=11 \rightarrow A. 11대$

Self-Consistency

Q. 주차장에 차가 3대 있다. 5분 뒤 3대의 차가
출고 있다. 5분 뒤 주차장에 차가 몇대?



Model ($m=3$)

1. 주차장에 3대, 5분 뒤 3대 들어옴,
· $3+5+3=11$

$a_1=11$

2. 지금 3대, 3대가 출고 있다,
 $3+3=6$

$a_2=6$

3. 3대가 이미 있다. 5분 뒤에 3대가 출가됨,
 $3+3=6$

$a_3=6$

$\arg\max_a \sum_{i=1}^3 \mathbb{1}(a_i=a) \Rightarrow a=6 \rightarrow A. 6대$

4. 핵심 결과

Table 2, 3 : 산술추론, 상식추론/기호추론 과제 성능 기록

Table2 : Arithmetic reasoning accuracy

	Method	AddSub	MultiArith	ASDiv	AQuA	SVAMP	GSM8K
	Previous SoTA	94.9^a	60.5 ^a	75.3 ^b	37.9 ^c	57.4 ^d	35 ^e / 55 ^g
UL2-20B	CoT-prompting	18.2	10.7	16.9	23.6	12.6	4.1
	Self-consistency	24.8 (+6.6)	15.0 (+4.3)	21.5 (+4.6)	26.9 (+3.3)	19.4 (+6.8)	7.3 (+3.2)
LaMDA-137B	CoT-prompting	52.9	51.8	49.0	17.7	38.9	17.1
	Self-consistency	63.5 (+10.6)	75.7 (+23.9)	58.2 (+9.2)	26.8 (+9.1)	53.3 (+14.4)	27.7 (+10.6)
PaLM-540B	CoT-prompting	91.9	94.7	74.0	35.8	79.0	56.5
	Self-consistency	93.7 (+1.8)	99.3 (+4.6)	81.9 (+7.9)	48.3 (+12.5)	86.6 (+7.6)	74.4 (+17.9)
GPT-3 Code-davinci-001	CoT-prompting	57.2	59.5	52.7	18.9	39.8	14.6
	Self-consistency	67.8 (+10.6)	82.7 (+23.2)	61.9 (+9.2)	25.6 (+6.7)	54.5 (+14.7)	23.4 (+8.8)
GPT-3 Code-davinci-002	CoT-prompting	89.4	96.2	80.1	39.8	75.8	60.1
	Self-consistency	91.6 (+2.2)	100.0 (+3.8)	87.8 (+7.6)	52.0 (+12.2)	86.8 (+11.0)	78.0 (+17.9)

Table3 : Commonsense and symbolic accuracy reasoning

	Method	CSQA	StrategyQA	ARC-e	ARC-c	Letter (4)	Coinflip (4)
	Previous SoTA	91.2^a	73.9 ^b	86.4 ^c	75.0 ^c	N/A	N/A
UL2-20B	CoT-prompting	51.4	53.3	61.6	42.9	0.0	50.4
	Self-consistency	55.7 (+4.3)	54.9 (+1.6)	69.8 (+8.2)	49.5 (+6.8)	0.0 (+0.0)	50.5 (+0.1)
LaMDA-137B	CoT-prompting	57.9	65.4	75.3	55.1	8.2	72.4
	Self-consistency	63.1 (+5.2)	67.8 (+2.4)	79.3 (+4.0)	59.8 (+4.7)	8.2 (+0.0)	73.5 (+1.1)
PaLM-540B	CoT-prompting	79.0	75.3	95.3	85.2	65.8	88.2
	Self-consistency	80.7 (+1.7)	81.6 (+6.3)	96.4 (+1.1)	88.7 (+3.5)	70.8 (+5.0)	91.2 (+3.0)
GPT-3 Code-davinci-001	CoT-prompting	46.6	56.7	63.1	43.1	7.8	71.4
	Self-consistency	54.9 (+8.3)	61.7 (+5.0)	72.1 (+9.0)	53.7 (+10.6)	10.0 (+2.2)	75.9 (+4.5)
GPT-3 Code-davinci-002	CoT-prompting	79.0	73.4	94.0	83.6	70.4	99.0
	Self-consistency	81.5 (+2.5)	79.8 (+6.4)	96.0 (+2.0)	87.5 (+3.9)	73.4 (+3.0)	99.5 (+0.5)

산술 추론 능력 향상

4개 모델 전부 추론 성능 향상

- 모델이 클수록 크게 향상됨
- SoTA 성능 달성

상식/기호 추론 능력 향상

6개 중 5개 과제에서 SoTA 달성

yes/no같은 과제는 성능 향상 더딴

성능의 원천은 표본 수가 아니라 **경로의 다양성**이다.

디코딩만 바꿔 성능을 향상시키는 실용적 방안이다.

5. 한계점

01 논문이 지적하는 한계

높은 계산 비용 : N개의 샘플 생성 = N배의 추론 비용 및 시간(Latency) 발생

- 5~10개 정도의 소수 경로를 대상으로 샘플링 하여 성능 향상 시도

02 전제의 한계

“오답은 제각각이고, 정답은 수렴한다”는 전제

모델이 잘못 학습하거나 함정에 빠질 경우 **‘잘못된 답’으로 수렴**

03 적용 과제의 한계

명확한 정답이 있는 수학·단답형·객관식 문제에만 유효

정답이 자유로운 과제(에세이, 코드 작성, 증명 등)는 '일관성(동점)' 정의 불가능

완화책으로 verifier 결합이나 프롬프트 다양화가 가능하지만, 이는 “추가 훈련 없음”이라는 장점을 포기하는 선택이다.

6. 추가연구

“다수결의 한계를 넘어서는 후속 연구”

01 과정 중심 검증 모델(Process Reward Model, PRM) 대표 연구: OpenAI *Let's Verify Step by Step* (Lightman et al., 2023)

추론의 각 단계별 정당성을 검증 모델로 점수를 매김

-> 중간 단계에서 오류가 감지되면 최종 답이 그럴듯해도 낮은 점수 부여

02 다중 에이전트 토론(Multi-Agent Debate) 대표 연구: *Improving Factuality and Reasoning through Multi-Agent Debate* (Du et al., 2023)

여러 에이전트가 서로의 추론 과정을 검토하고 논박

-> 한 에이전트가 편향된 전제에 빠져도 다른 에이전트가 지적 가능

03 프롬프트 다양화를 통한 접근

“역발상으로 접근하라”, “반비례를 먼저 찾아라” 등 문제 입력 각도 자체를 다변화하여 샘플링

-> 특정 프롬프트 템플릿이나 질문 방식이 유도하는 고질적 편향 상쇄

RESEARCH SEMINAR · 2026

감사합니다

머신러닝 특론 | 202411970 정시윤

2026.09.08