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

Jason Wei et al. (Google Research, Brain Team) - 2022

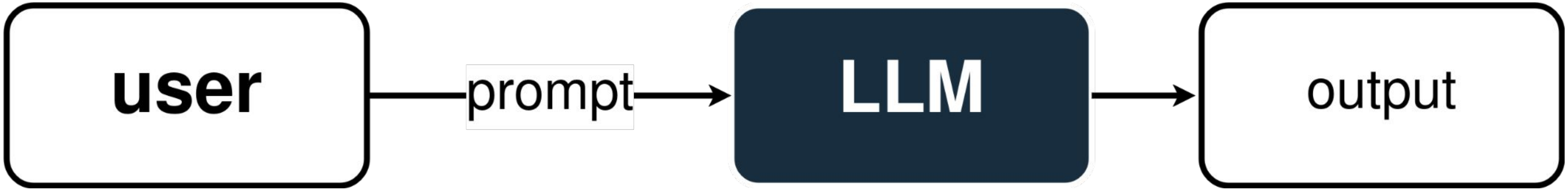
presentation by Alexander Wehner
Reasoning with LLMs seminar
summer semester 2025
UdS

Structure of this talk

- Introduction & basic terminology
- Prompting Style Guide
  - Basic Prompting
  - Few-Shot Prompting
  - Chain-of-Thought Prompting
- Performance Review of Chain-of-Thought Prompting
  - Arithmetic Reasoning
  - Common Sense Reasoning
  - Symbolic Reasoning
- Robustness
- FAQ
- Discussion

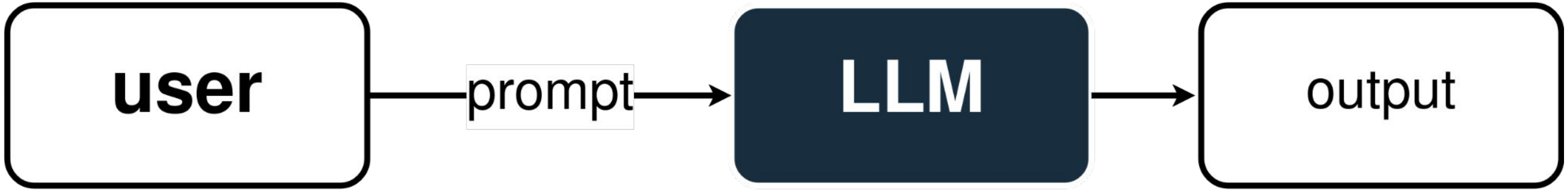
Large Language Models

- for the moment, imagine the LLM as a black box
- produces text output when given a text input prompt
- many architectures and types of LLMs (details don’t matter right now)



- LLMs have varying numbers of parameters (typically in the billions)
- more parameters = more powerful / capable
- usually called “model size” or “model scale” in literature

What is prompting?



Example prompt for a reasoning task:

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

What is reasoning?

decompose the problem into intermediate steps and apply logic:

- start with 23 apples
- subtract 20
- 3 apples remain
- add 6 apples
- now we have 9

derive the solution: 23 - 20 + 6 = 9

We want LLMs to do this, or something analogous.

Benchmarks

- benchmarking = measuring performance of models
- done with standardized collections of specific tasks for LLMs to solve

Examples:

- Arithmetic: GSM8K, SVAMP, AQuA, MAWPS, etc.
- Common Sense: CSQA, StrategyQA, SayCan, etc
- noteworthy: BIG-bench
    - very large collection of benchmarking tasks
    - covers a vast range of different types of tasks and domains

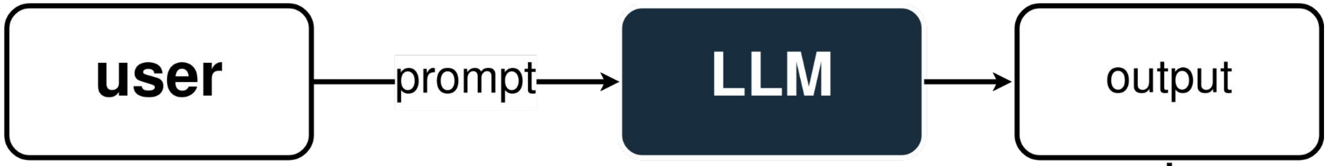
Prompting Style Guide

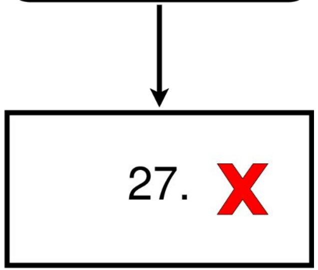
The ones we care about right now:

- Zero-Shot prompting (aka ‘normal’ prompting)
- Few-Shot prompting (One-Shot, Two-Shot, or more)
- Chain-of-Thought prompting

Zero-Shot prompting

Zero-Shot prompting is just giving the model your prompt unaltered:





Example:

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

Problems:

- in many cases, performance is not optimal
- does not utilize the full reasoning potential of most models
- in case of mistakes, finding out what went wrong is hard

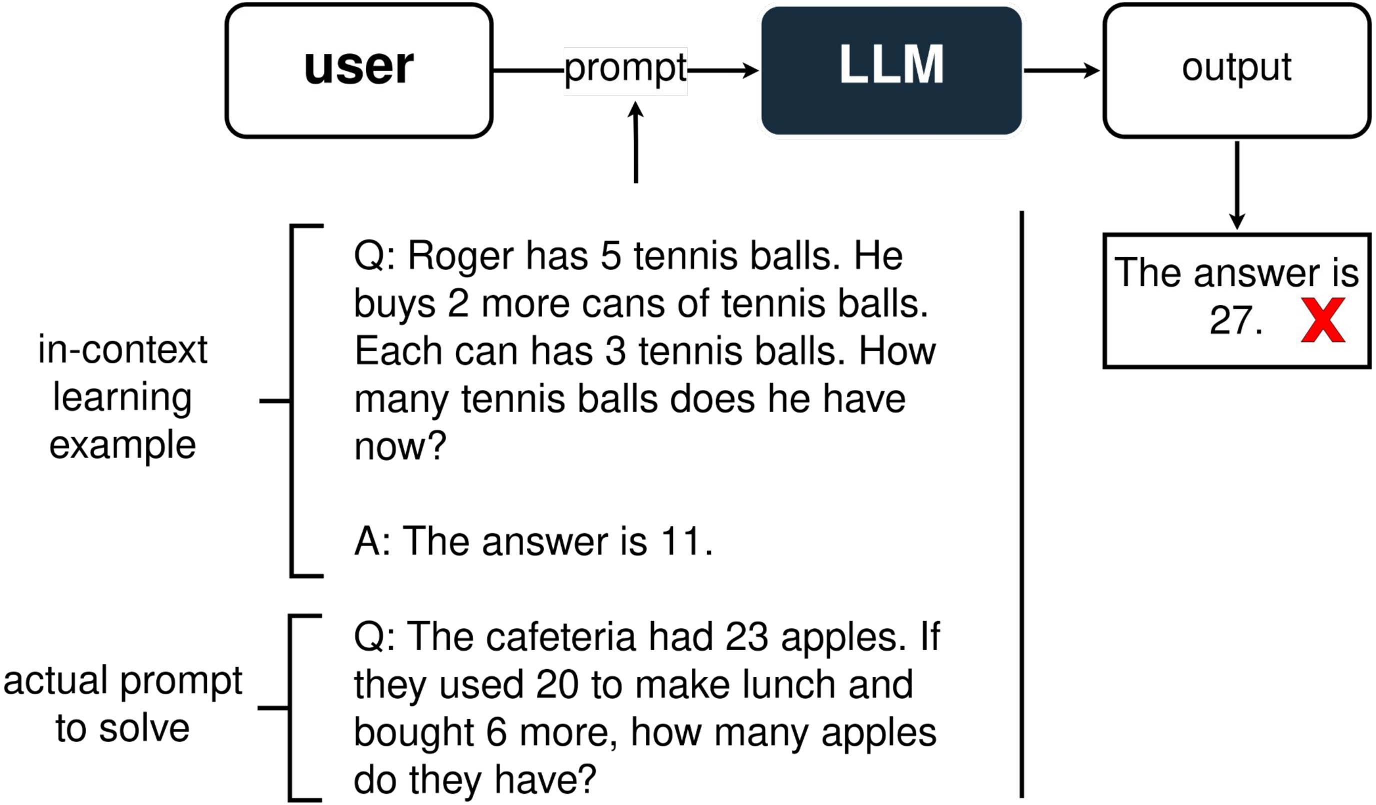
Few-Shot prompting

improves on Zero-Shot prompting in a distinct way:

In-context learning

- enables LLMs to generalize to (learn from) examples given to it in the input context
- examples are given as an input-output pair, together with the prompt
- examples demonstrate how a similar task would be solved correctly

Example for Few-Shot prompting:



Pros & Cons of Few-Shot

Pros:

- adequate boost to model performance on most benchmarks
- no finetuning needed
- in-context learning examples can be adapted to match the desired task

Cons:

- does not make use of natural language rationales to boost reasoning
- sensitive to example quality
- it’s still hard to understand what went wrong, in case of mistakes

Chain-of-Thought Prompting

further improves Few-Shot prompting by:

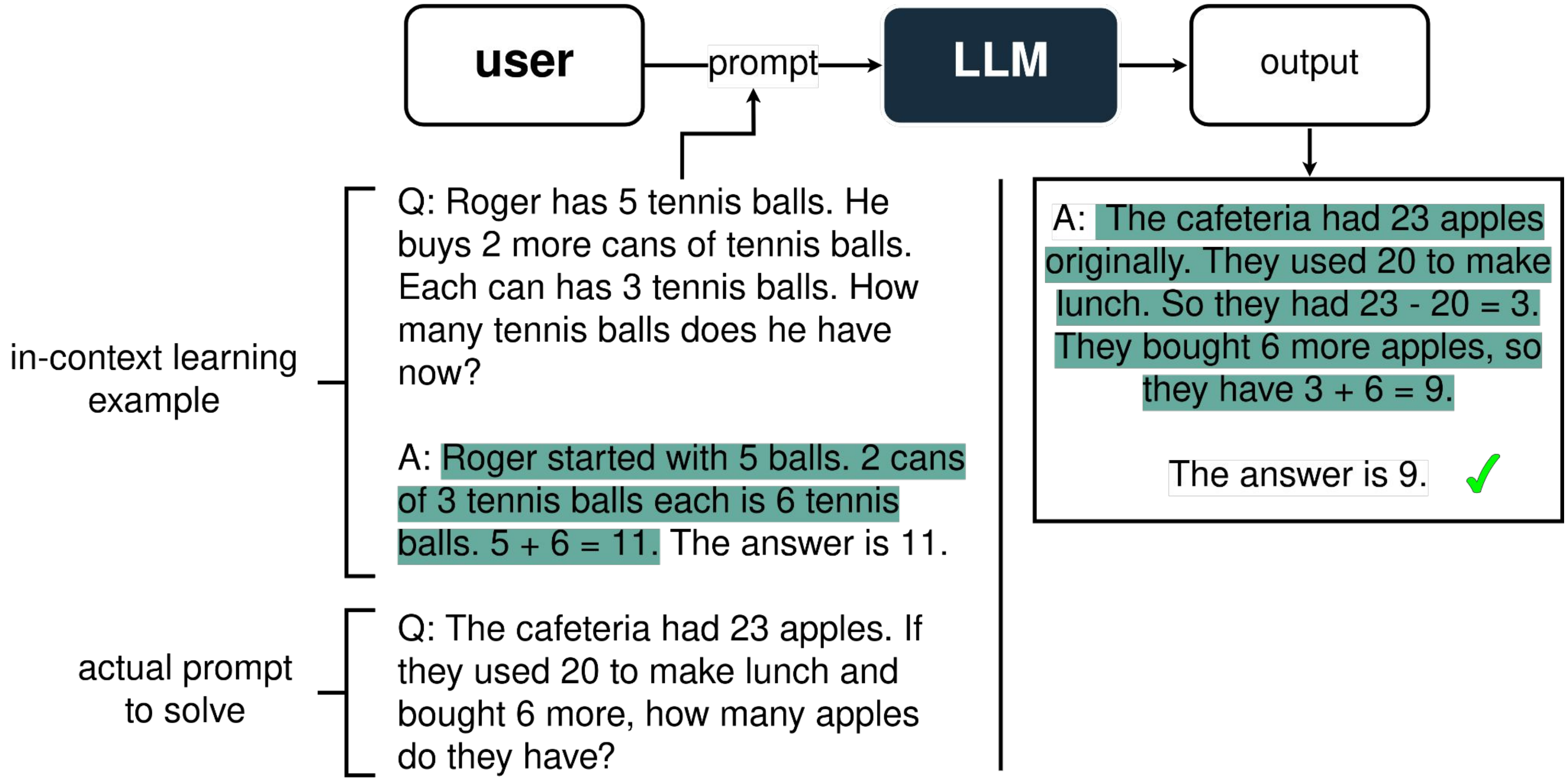
- utilizing natural language rationales (chains of thought) to boost reasoning ability
- inducing step-by-step reasoning while generating the output

Similar to Few-Shot prompting, but:

- in-context learning examples now contain additional chains of thought
- models can learn to apply natural language rationales to the task

⇒ the model “reasons” as it generates

Example for Chain-of-Thought prompting:



Pros & Cons of Chain-of-Thought Prompting

Pros:

- significant boost to model performance on most benchmarks
- makes use of natural language rationales to improve reasoning ability
- retains and builds on the in-context learning feature of Few-Shot prompting

Cons:

- chains of thought will have to be manually added to prompts
- Spoiler: performance boost is directly tied to model scale (more on that later)

Performance Review of Chain-of-Thought Prompting

Setup:

- 5 large language models in total were tested:
  - GPT-3 (350M, 1.3B, 6.7B, 175B)
  - LaMDA (420M, 2B, 8B, 68B, 137B)
  - PaLM (8B, 62B, 540B)
  - Codex
  - UL2 (20B)
- 3 types of tasks were evaluated on various benchmarks:
  - Arithmetic reasoning (GSM8K, SVAMP, ASDiv, AQuA, MAWPS)
  - Commonsense reasoning (CSQA, StrategyQA, SayCan, 2 subsets of BIG-bench)
  - Symbolic reasoning (2 customized toy tasks)

Performance Review of Chain-of-Thought Prompting

Setup:

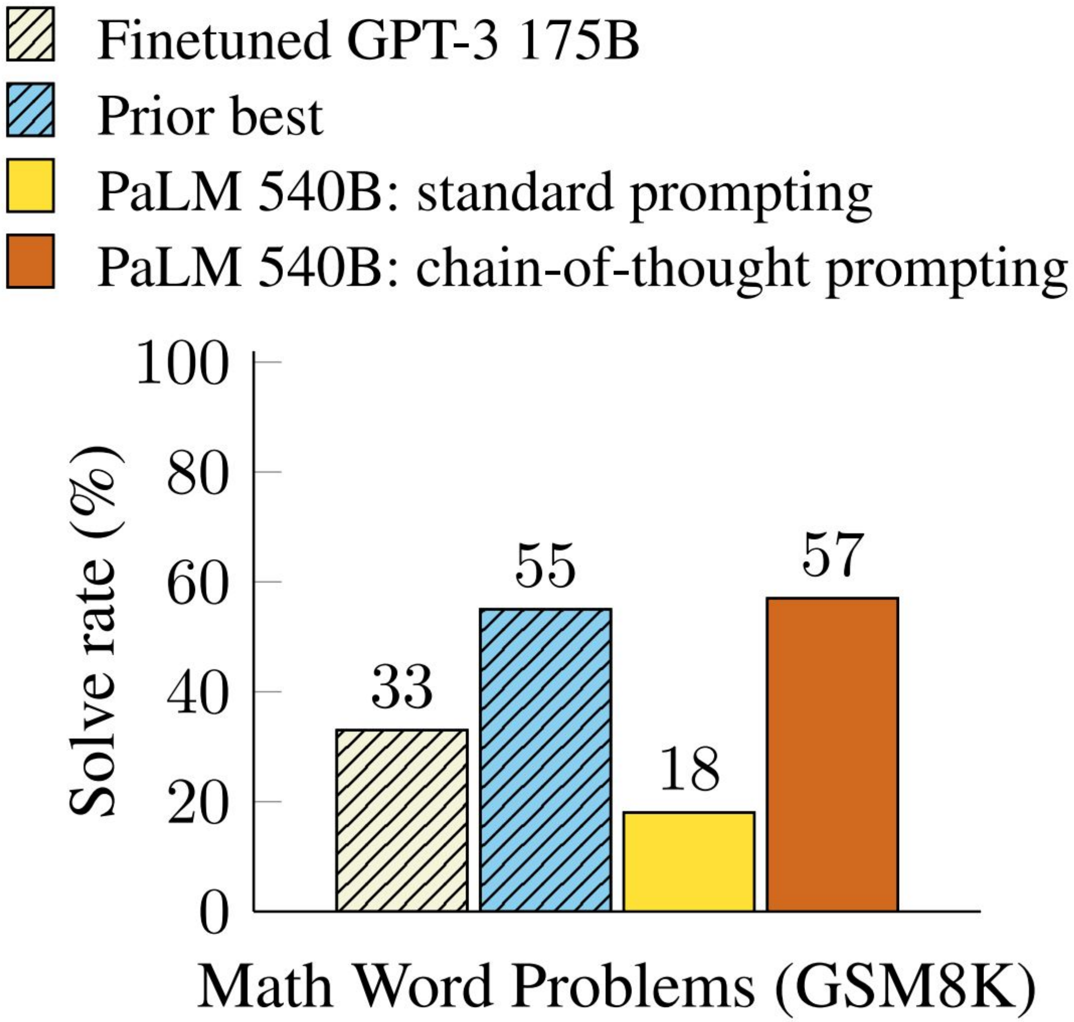
- prompts in 2 different styles were used:
    - Few-Shot (as baseline)
    - Chain-of-Thought

- custom Few-Shot examples with added chains of thought were composed to create the Chain-of-Thought prompts

General performance:

on GSM8K, PaLM 540B using Chain-of-Thought prompting achieved:

- nearly double the performance of a finetuned GPT-3 (175B)
- new state of the art performance
- triple the performance of standard Few-Shot prompting



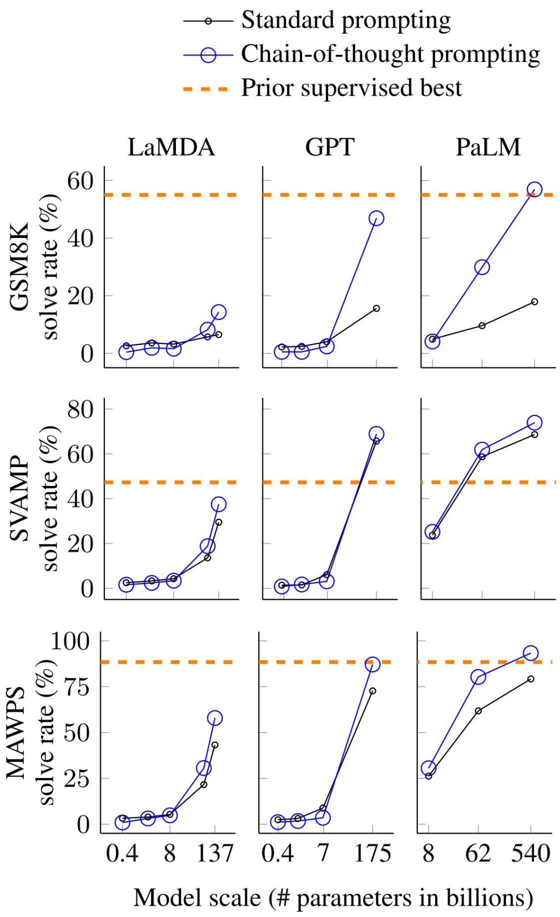
Arithmetic Reasoning Performance

Key takeaways:

- new state of the art performance (PaLM 540B, on 3 of the benchmarks)

but:

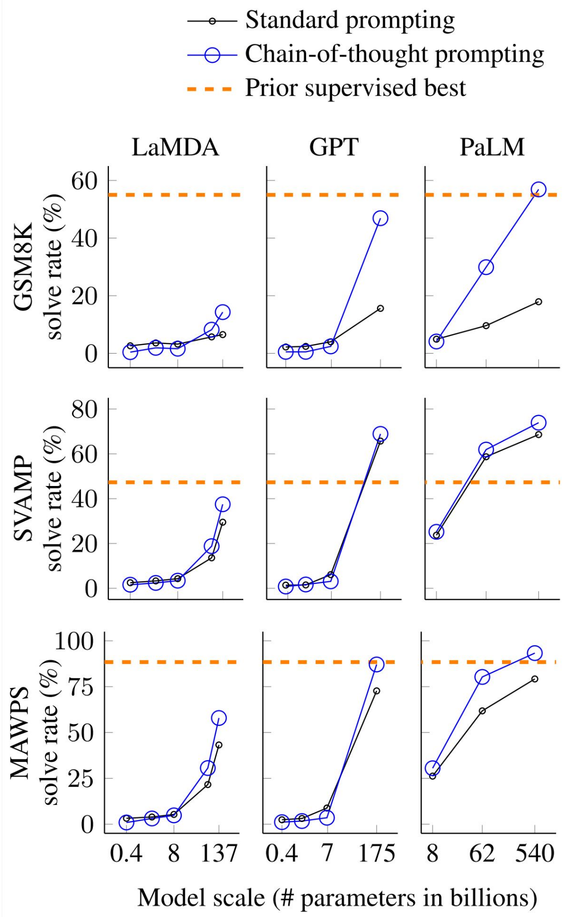
- smaller models don’t seem to benefit much
- better performance gains for harder problems (like GSM8K)



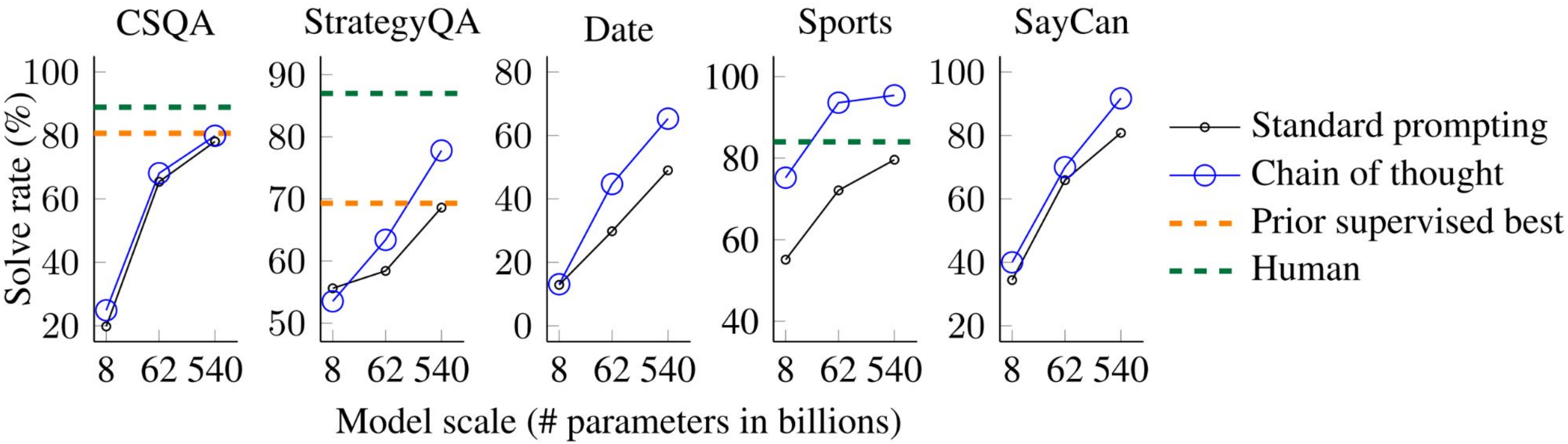
Arithmetic Reasoning Performance

this suggests that Chain-of-Thought…

- … is an emergent ability of model scale, only boosting performance of larger models (~100B or more)
- … can at times even hold back smaller models
- … does not help on easy tasks

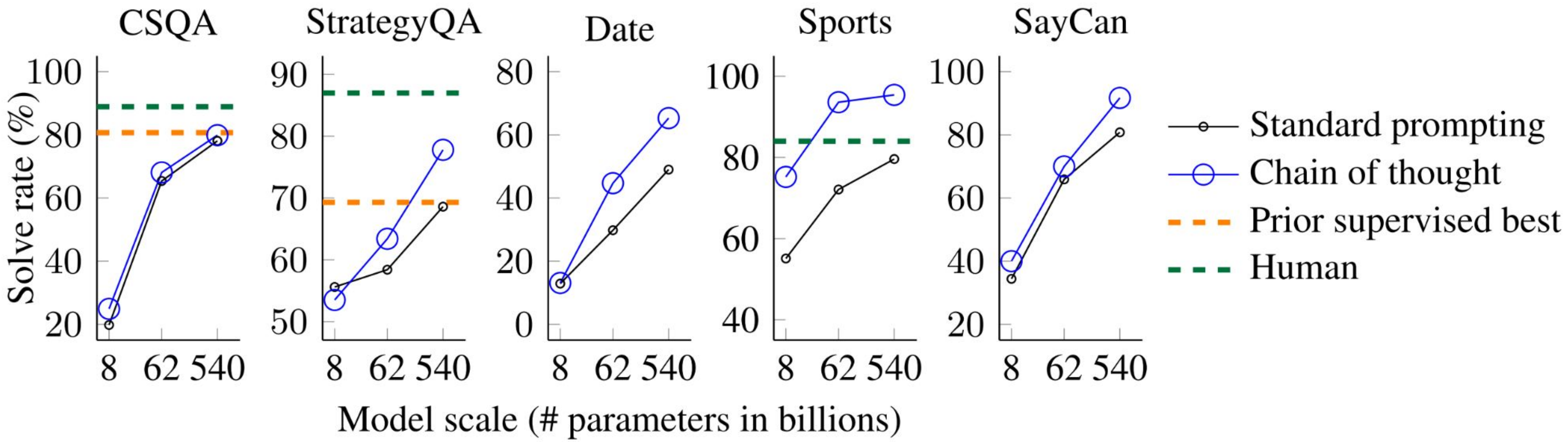


Commonsense Reasoning Performance



- PaLM 540B outperforms prior state of the art on StrategyQA
- scaling up model size improves the baseline
- Chain-of-Thought further boosts performance
- effect is most prominent on the largest model (PaLM 540B)

Commonsense Reasoning Performance



this means:

- Chain-of-Thought can help with commonsense reasoning
- similar or less performance gains, compared to arithmetic reasoning results

Symbolic Reasoning Performance

2 customized toy tasks:

- last letter concatenation
  - concatenate the last letters of words

    “French Toast” ⇨ “ht”

- coin flip
  - track the state of a coin during a series of actions (flipping)

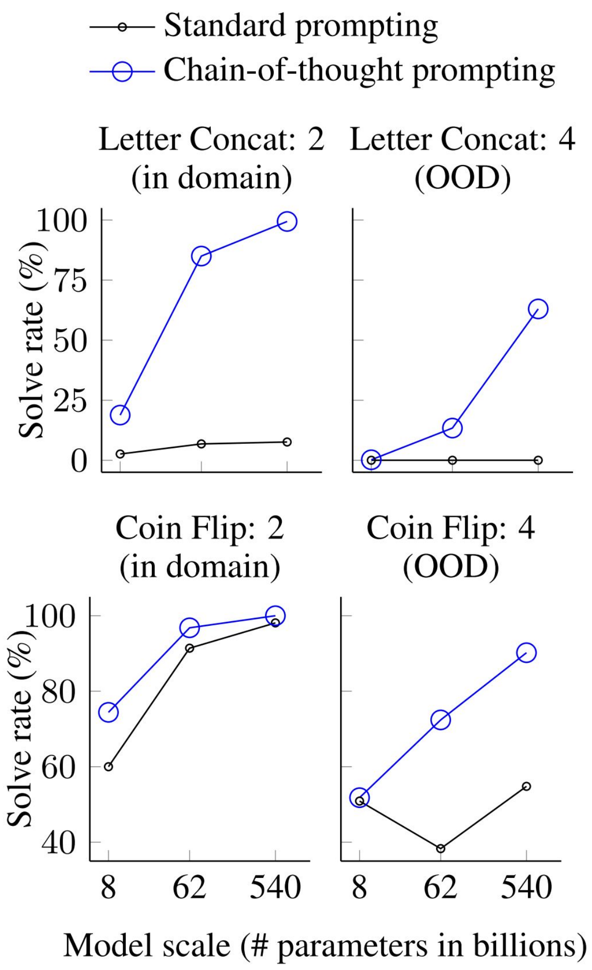
2 different types of in-context learning examples were tested:

- with the same number of steps as the task (in-domain)
- with a different number of steps than the task (out-of-domain or OOD)

Symbolic Reasoning Performance

in-domain results:

- with Chain-of-Thought, paLM 540B reaches almost 100% solve rate
- for last letter concatenation, baseline prompting fails across all model sizes
- note that baseline PaLM 540B already solves coin flip

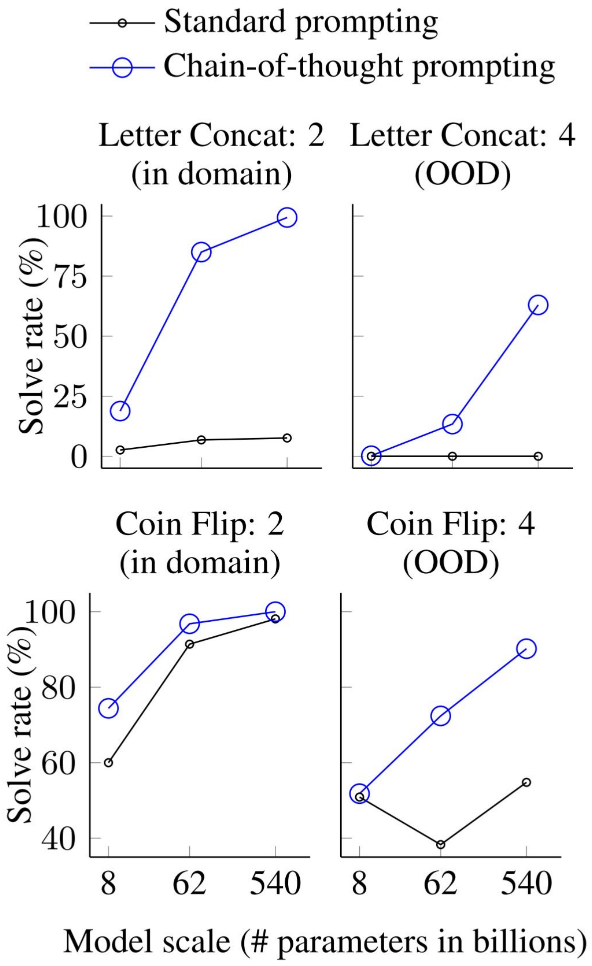


Symbolic Reasoning Performance

out-of-domain results:

- standard Few-Shot prompting fails completely

- Chain-of-Thought prompting boosts performance, but still lower than in-domain performance

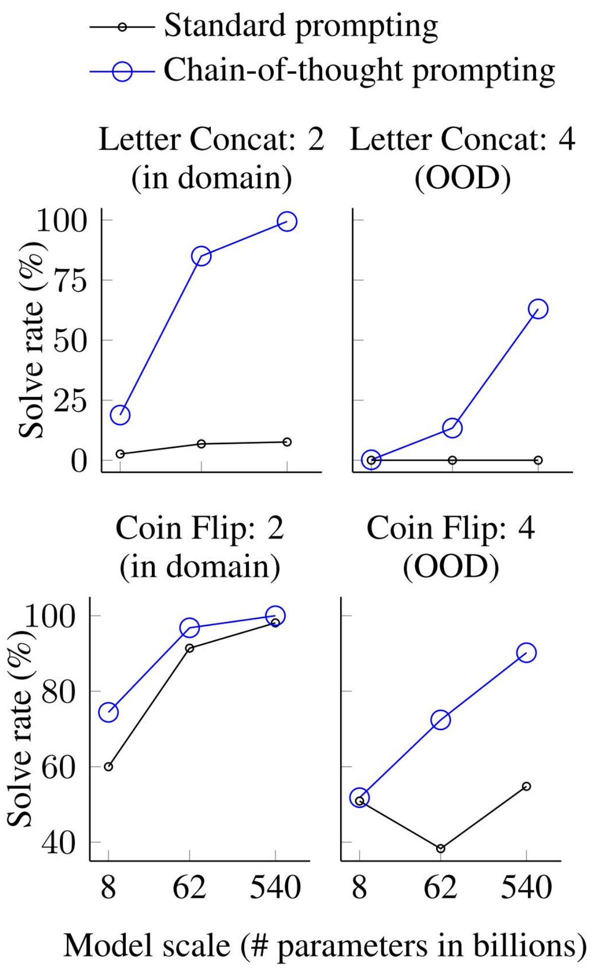


Symbolic Reasoning Performance

key takeaways:

- Chain-of-Thought facilitates length generalization beyond seen chains of thought (as seen in OOD results)

- small models still fail, even with Chain-of-Thought prompting (emergent ability)



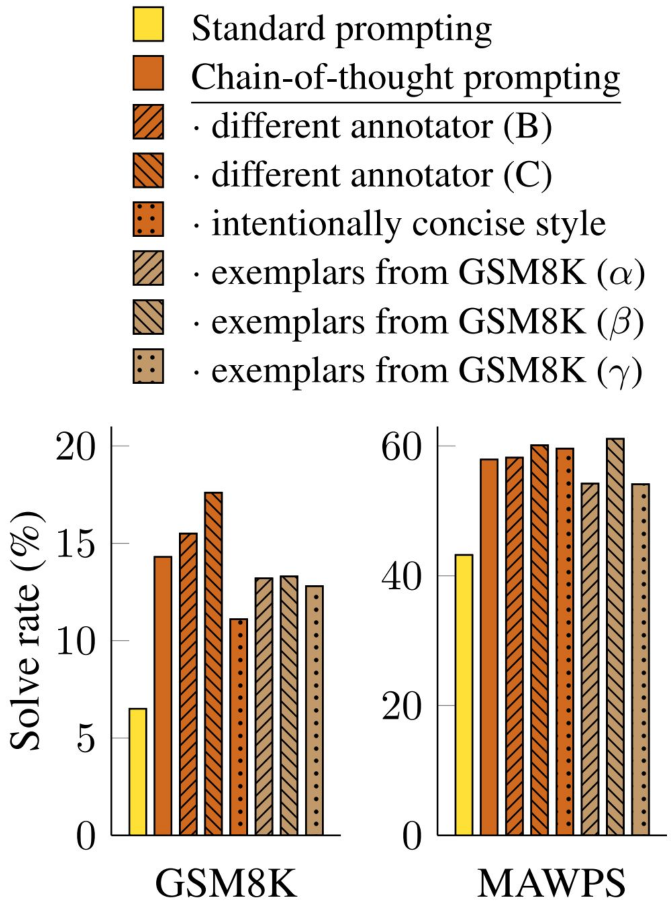
Robustness

performance data for different permutations of chains of thought shows:

- robustness to different annotators
- robustness to linguistic style & wording
- robustness to different examples

note:

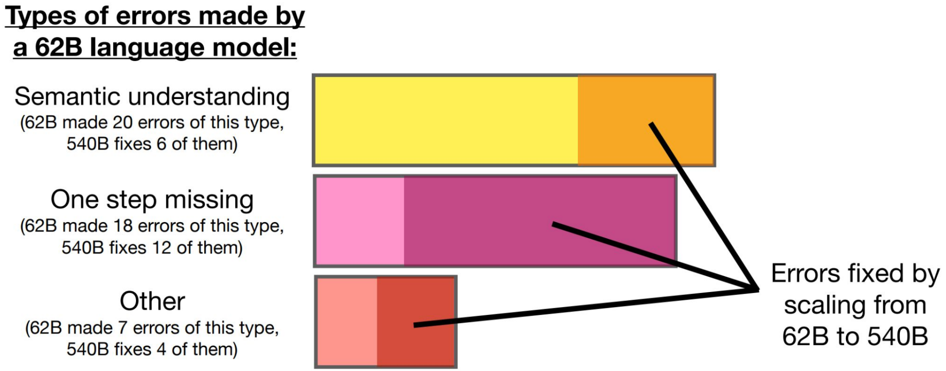
- variance among different permutations is expected (in-context learning is sensitive to example quality)
- they all outperform the baseline by a large margin



FAQ

Why does increasing model scale improve Chain-of-Thought prompting?

- smaller models tend to make mistakes in the chains of thought they generate
- most can be fixed by increasing scale
- small models seem to not have enough inherent reasoning ability that can benefit from Chain-of-Thought
- by increasing scale, more reasoning ability emerges → better chains of thought → even better reasoning



FAQ

Will Chain-of-Thought prompting improve performance for any task?

Can I apply it to everything?

in theory: maybe yes

in practice: it depends on the task

- task hardness plays an important role
- non-reasoning tasks (e.g. creative writing or coding) will likely not benefit much

Recap:

Chain-of-Thought is …

- … an emergent ability (needs ~100B parameters or more)
- … effective in boosting reasoning performance on reasoning tasks
- … applicable to off-the-shelf LLMs, no finetuning needed
- … somewhat easy to implement, but manually tedious in practice
- … limited not only by model scale but also task hardness
- … possibly also limited by task domain (non-reasoning / creative tasks)

Open questions:

- How can we know if the model really is “reasoning”?
  - - We don’t know that (yet). Also, that’s a very philosophical question.
- Can we guarantee correct reasoning paths?
  - - No, but for now increasing model scale seems to help.

A quick look into the future

>> **Large Language Models are Zero-Shot Reasoners - Kojima et al., 2022** <<

Simply adding the string

“Let’s think step by step”

to a Zero-Shot prompt has a very similar effect.

This does not require in-context learning examples.

They called this **Zero-Shot-Chain-of-Thought.**

Thank you for your attention.

References:

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

   Wei et al., 2022 - https://arxiv.org/abs/2201.11903

2. Large Language Models are Zero-Shot Reasoners

   Kojima et al., 2022 - https://arxiv.org/abs/2205.11916