263-5354-00L
Large Language Models

Adapted for CSI 5386 at the University of Ottawa, 2024

Prompting and Zero-shot inference

or Few shot learning

Mrinmaya Sachan
msachan@ethz.ch

Lecture Outline

1. What is Prompting? How do we prompt language models?

2. Prompt Engineering
   - Manual prompts
   - Automated prompts
   - Discrete vs continuous prompts

3. Chain of thought prompting

4. In-context learning

A Short History of Language Modelling

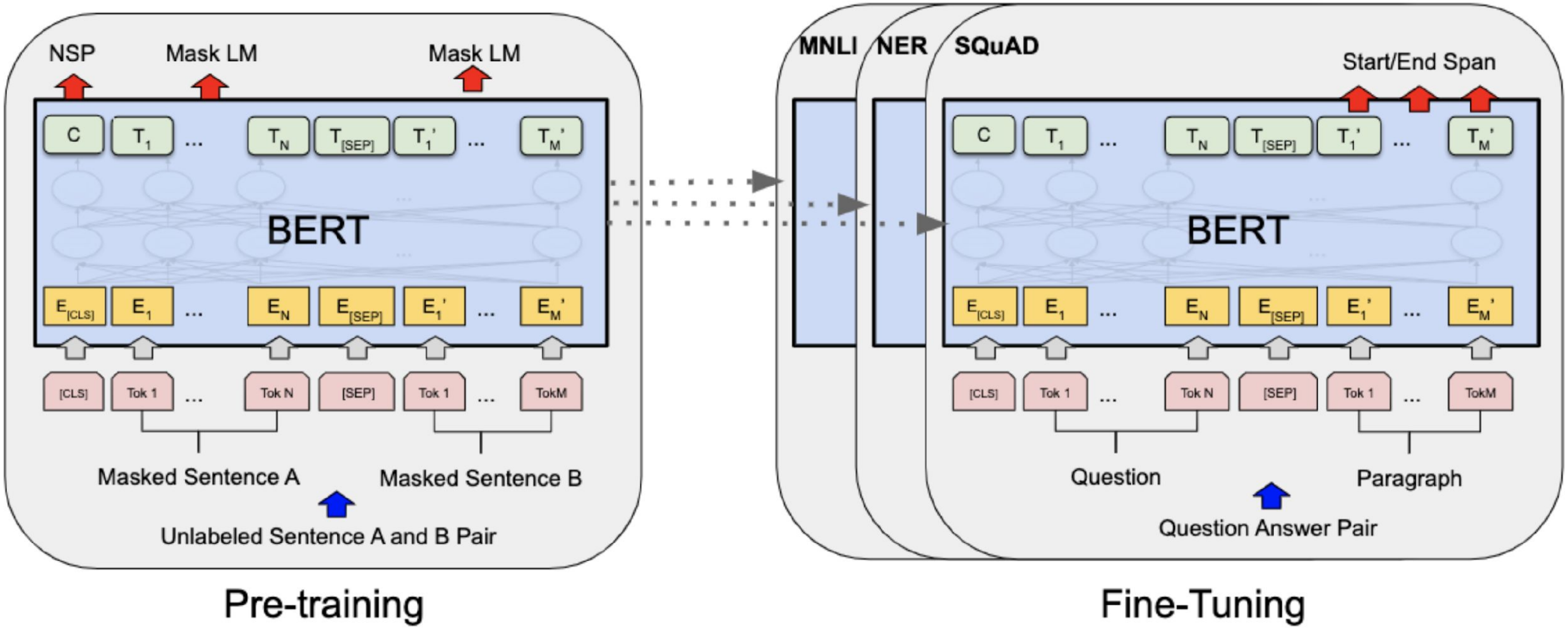
- Language models date way back to Shannon, etc. in the 50s-60s

- **Until 10-15 years ago:** Language models used a database of **word counts** from a corpus of text to estimate probabilities
  - Improved speech recognition and machine translation systems
  - NLP systems for other tasks ignored language models and required **millions of examples** to learn tasks

- **10 years ago:** Deep Learning coupled with **Transfer learning** made language models effective
  - NLP systems began to use language models as a starting point to learn tasks, but still need **thousands** of examples to do so

- **Last 3 years: Scale up** (data & model) and **prompting**
  - GPT series of models
  - **Simply prompt or instruct these models to do the task**

Earlier in this class: Supervised Finetuning



Typically need **tens of thousands** of examples!!!

Does not work well if we no not have finetuning data …

- The model might not be able to adapt to the finetuning data based on just a small dataset and might forget everything it has learner during pretraining.

Parameter efficient finetuning

A partial solution to this problem:

- **Just finetune a subset of the parameters for each task**

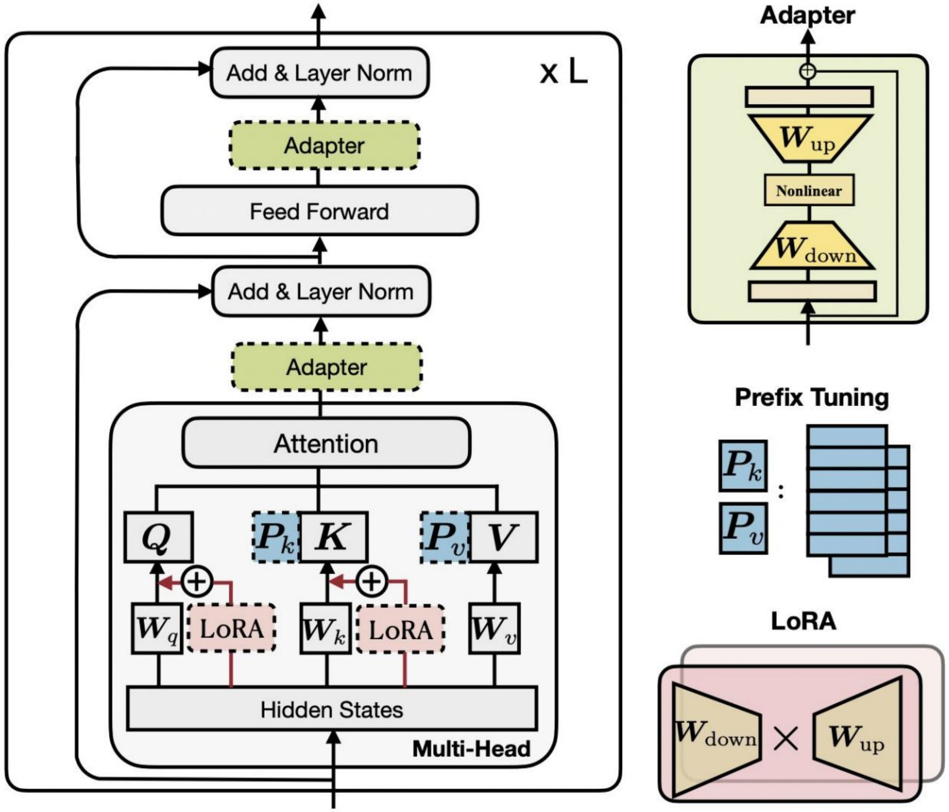


Image: (He et al. 2022)

Still need **thousands** of examples!!!

Prompting: Why even finetune?

For many tasks, **supervised finetuning data may not be available**

Key idea: Prompting enables models to circumvent this by learning a LM that models the probability P(x; θ) of the input text x, and using this probability to predict y, reducing or obviating the need for large supervised datasets.

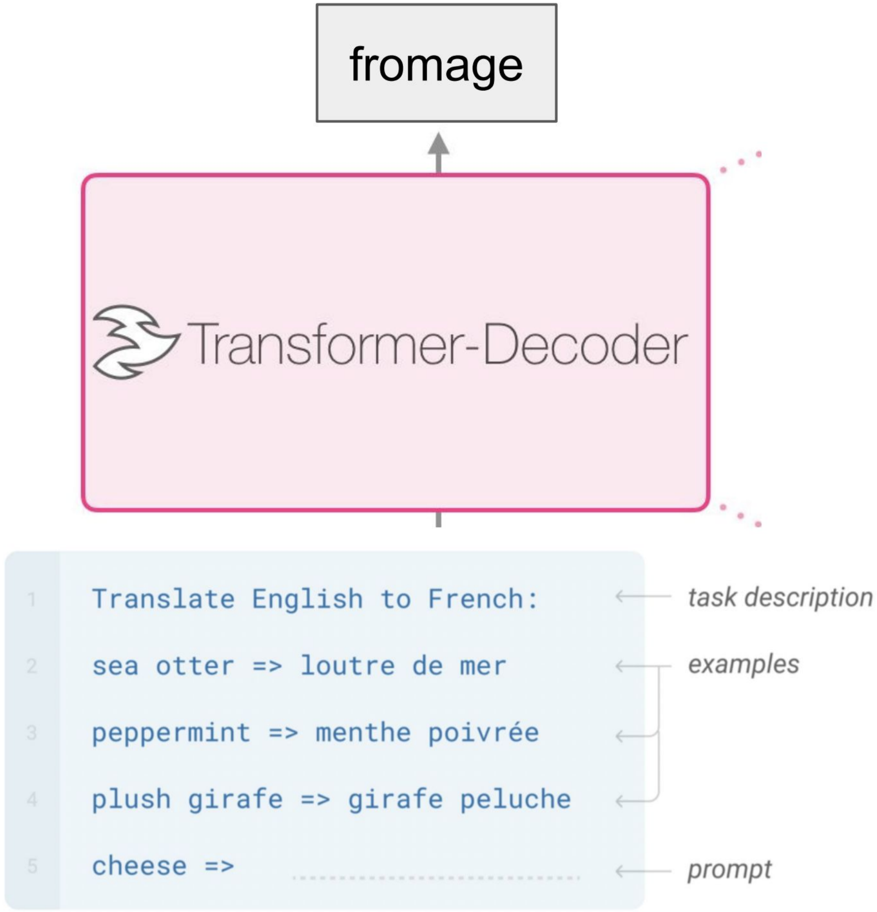
Prompting is **non-invasive**:

- It does not introduce any additional parameters or require direct inspection of a model's representations.

It can be thought of as a **lower bound on what the model "knows" about the new task (x □ y)** and this information is simply extract from the LM via prompting.

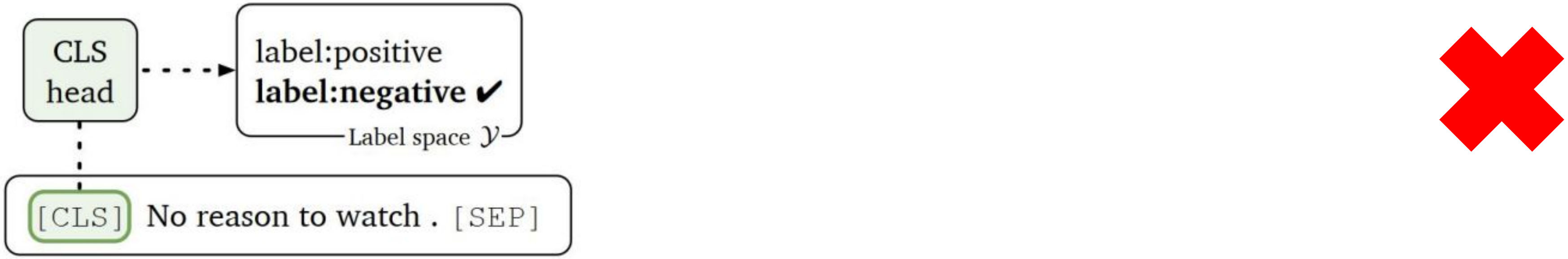
Key idea: We manually design a “prompt” that demonstrates how to formulate a task as a generation task.

No need to update the model weights at all!

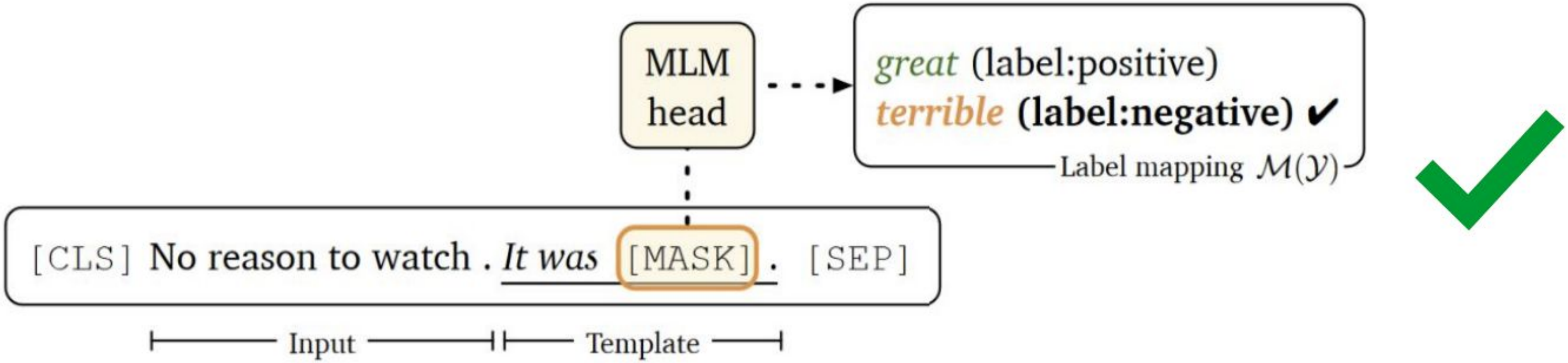


```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

Head-based fine-tuning



Prompting:


MLM head
great (label:positive)
terrible **(label:negative)** ✔
Label mapping $\mathcal{M}(\mathcal{Y})$
[CLS] No reason to watch . *It was* [MASK] . [SEP]
Input
Template
✓

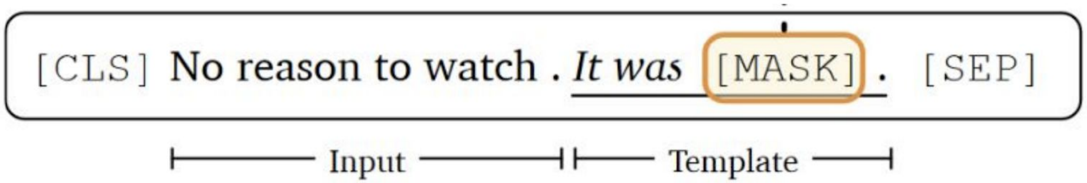
We can also do prompt-based fine-tuning if we have supervised data.

Prompting a bit more formally

Lets say we have a **classification** task, e.g. sentiment classification

Input: x_1 = No reason to watch.

Step 1. Formulate the downstream task into a (Masked) LM problem using a *template:*

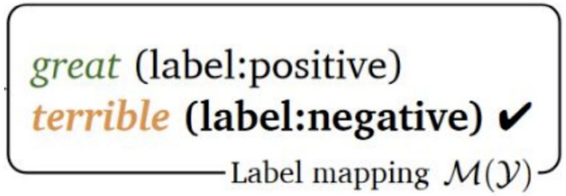


[CLS] No reason to watch . *It was* [MASK] . [SEP]

Input

Template

Step 2. Choose a *label word mapping* $\mathcal{M}$, which maps task labels to individual words.



great (label:positive)

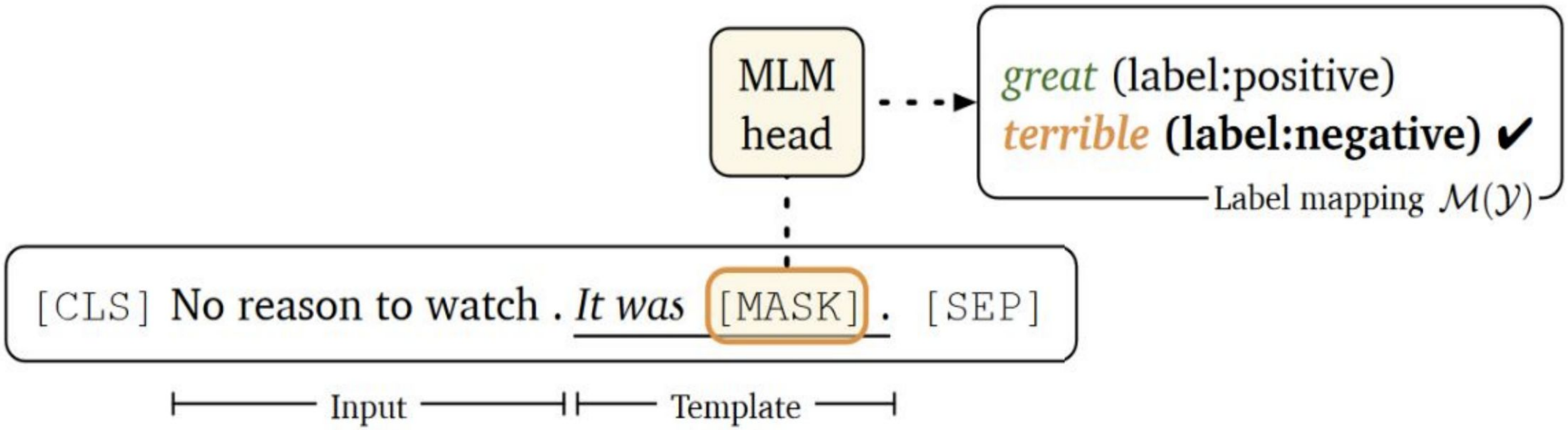
terrible **(label:negative)** ✔

Label mapping $\mathcal{M}(\mathcal{Y})$

Prompting a bit more formally

Step 3. ~~Fine tune~~ Directly use the LM to fill in the correct label word.

$$
\begin{aligned}
p(y \mid x_{\text{in}}) &= p\left(\texttt{[MASK]} = \mathcal{M}(y) \mid x_{\text{prompt}}\right) \\
&= \frac{\exp\left(\mathbf{w}_{\mathcal{M}(y)} \cdot \mathbf{h}_{\texttt{[MASK]}}\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\mathbf{w}_{\mathcal{M}(y')} \cdot \mathbf{h}_{\texttt{[MASK]}}\right)},
\end{aligned}
$$


MLM head
great (label:positive)
terrible (label:negative) ✔
Label mapping $\mathcal{M}(\mathcal{Y})$
[CLS] No reason to watch . It was [MASK] . [SEP]
Input
Template

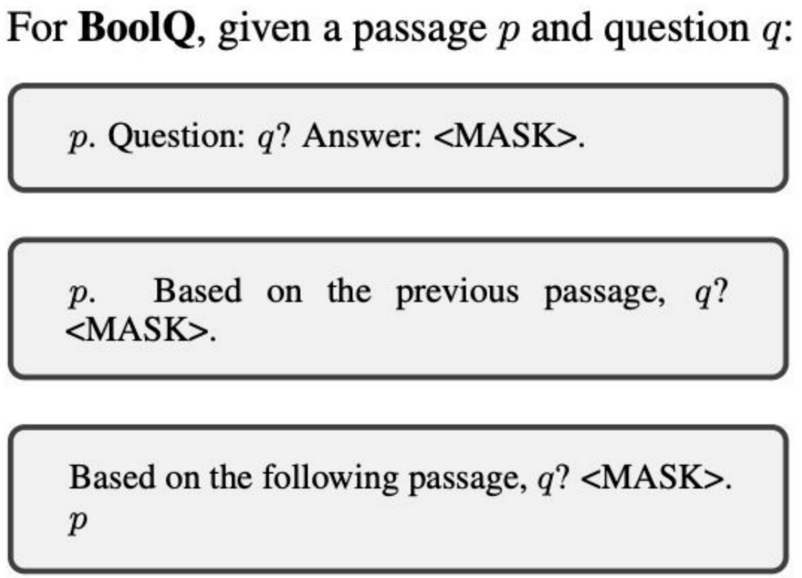
What makes a good prompt?
for an NLP task,

GPT3: "a good prompt is one that is specific and ***provides enough context for the model*** *to be able to generate a response that is relevant to the task."*

The Dark Art of Prompt Engineering

Question Answering:

BoolQ: given a passage q and question p, design a prompt for question answering



For **BoolQ**, given a passage p and question q:

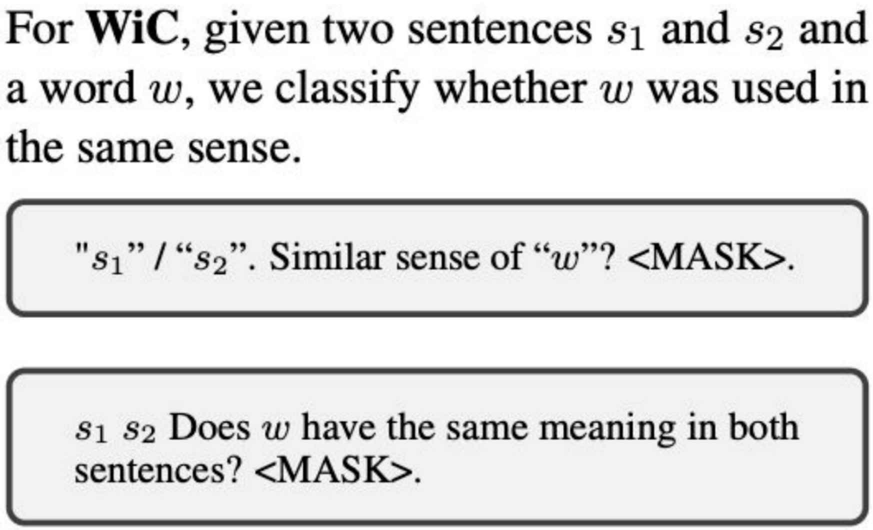
p. Question: q? Answer: <MASK>.

p. Based on the previous passage, q? <MASK>. ✓

Based on the following passage, q? <MASK>. p

Word sense:

WiC: given two sentences S1 and S2, and a word W, design a prompt to determine whether W was used in the same sense in both sentences.



For **WiC**, given two sentences s_1 and s_2 and a word w, we classify whether w was used in the same sense.

"s_1" / "s_2". Similar sense of "w"? <MASK>.

s_1 s_2 Does w have the same meaning in both sentences? <MASK>. ✓

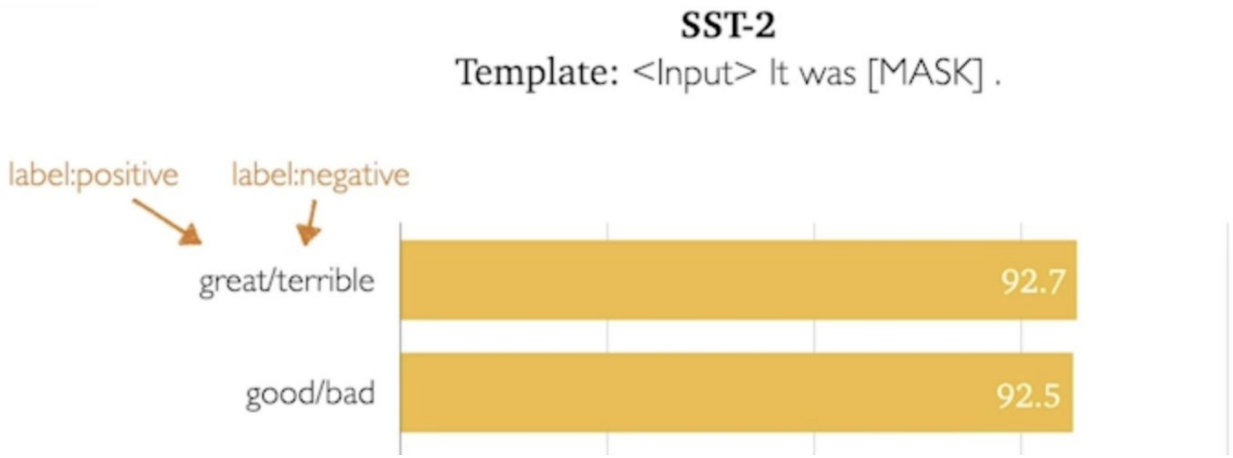
The Dark Art of Prompt Engineering

In general, prompting is a **dark art**

- Requires **domain expertise and trial and error**

The challenge is to find a template T and label words M(y) that work in conjunction

- Slight variations in prompts can lead to differences!



Can we Automate Prompt Design?

Some initial work on automating discrete prompt design with moderate success:

1. **Mine prompt candidates from a large corpus (Jiang et al. 2020)**

- Search for strings in a large text corpus, e.g. Wikipedia, that contain both training inputs x and outputs y
- Identify the middle words or dependency paths between them
- Use the middle words/paths as templates in the form of "`[X]` middle words `[Z]`"

2. **Paraphrase approach**

- Translating the prompt into another language and back (Jiang et al., 2020)
- Use a thesaurus to replace words (Yuan et al., 2021)
- Train a neural prompt rewriter designed/trained with the objective of improving the accuracy of systems that use the prompt (Haviv et al., 2021)

3. **Training a text generation model for generating prompts**

- Pre-train a T5 model for the template search

Generate multiple prompts from all the training examples, or a few well-chosen examples.

Automating Prompt Design Somewhat Works

Recall the LAMA probe.

Manually designed prompts in LAMA probe

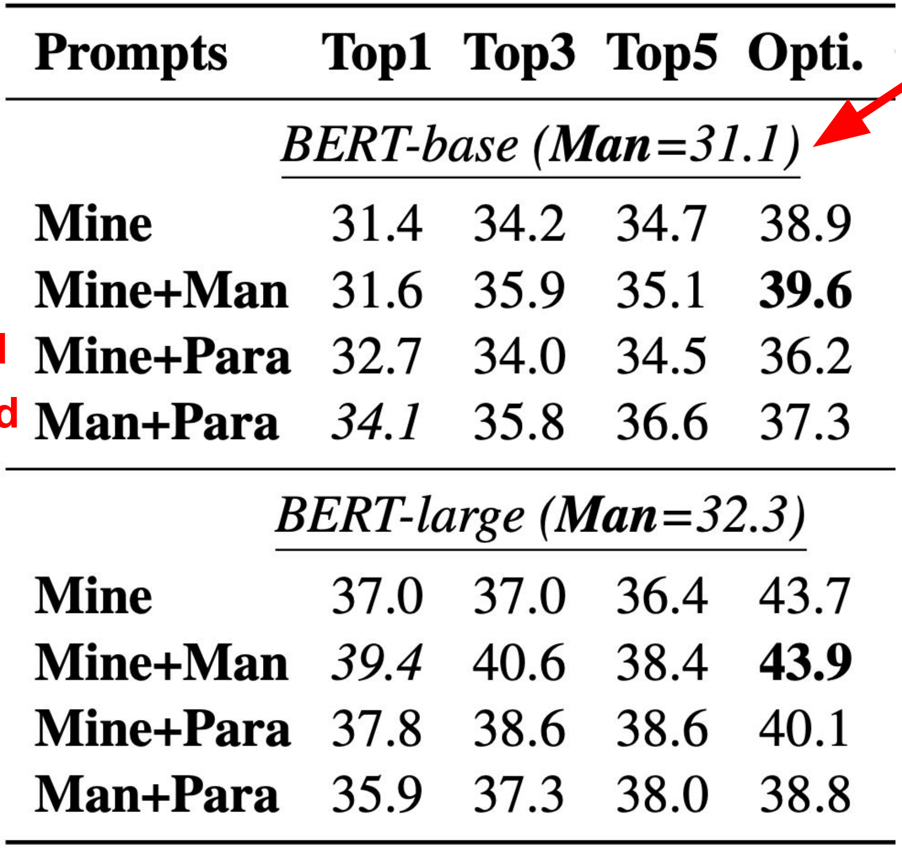
LAnguage Model Analysis (LAMA): A set of knowledge sources (set of facts) for analyzing the factual and commonsense knowledge contained in LLMs (Perrori et al., 2019)

Mined prompts

Mined+Manual

Mined□Paraphrased

Manual□Paraphrased



Prompts	Top1	Top3	Top5	Opti.
	*BERT-base (**Man**=31.1)*			
Mine	31.4	34.2	34.7	38.9
Mine+Man	31.6	35.9	35.1	**39.6**
Mine+Para	32.7	34.0	34.5	36.2
Man+Para	*34.1*	35.8	36.6	37.3
	*BERT-large (**Man**=32.3)*			
Mine	37.0	37.0	36.4	43.7
Mine+Man	*39.4*	40.6	38.4	**43.9**
Mine+Para	37.8	38.6	38.6	40.1
Man+Para	35.9	37.3	38.0	38.8

(Jiang et al. 2020)

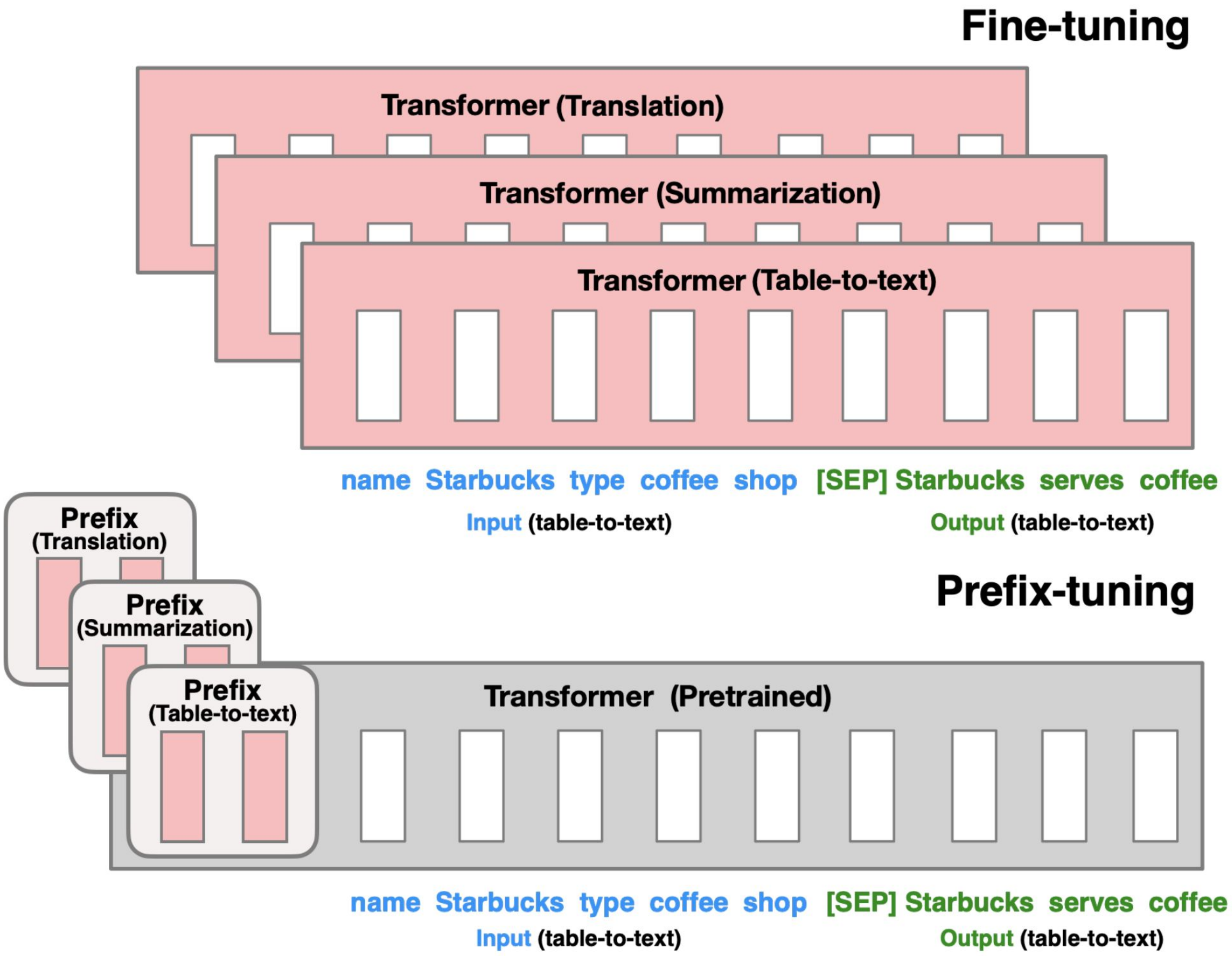
Opti. Prompt outputs are ensembled (i.e predictions are weighted and combined)

Continuous Prompts

Recent approaches have explored continuous prompts (also sometimes known as soft prompts) that prompt the model directly in its embedding space.

One approach (we have already seen) is **prefix tuning**.

Prefix Tuning (Li et al. 2021)


Fine-tuning
Transformer (Translation)
Transformer (Summarization)
Transformer (Table-to-text)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)
Prefix-tuning
Prefix (Translation)
Prefix (Summarization)
Prefix (Table-to-text)
Transformer (Pretrained)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)

Prefix-tuning freezes Transformer parameters and **only optimizes the prefix** (red prefix blocks).

Advanced Prompting

1. Prompting with demonstrations

2. Chain of Thought Prompting

Prompting with Demonstrations

Lets assume we have a **few shots**:

- We have a **small collection of input-out exemplars**.
- These exemplars serve as demonstrations of the behavior that one would like the LM to emulate.

We can use prompting in this case:

Key idea: Augment the standard prompt "France's capital is `[X]`" by **prepending the examples**:

"Great Britain's capital is London . Germany's capital is Berlin . France's capital is `[X]`".

No parameter updates to the model as before!

Challenges in Prompting with Demonstrations

Prompting with demonstrations for few-shot learning is very popular.
It works!

However, as in prompt design, we have **several questions**:

1. What examples to include in the prompt to make the demonstration effective?
2. How to order the examples in the prompt. Different demonstration orders can result in very different performance (Lu et al., 2021).

We can partially tackle this issue, e.g, by:

1. using sentence embeddings to **sample examples that are close to the input in the embedding space** (Liu et al., 2021a; Drori et al., 2022).
2. As for the order of the labeled examples, we can also learn a model to **score different candidate permutations** (Lu et al., 2021).

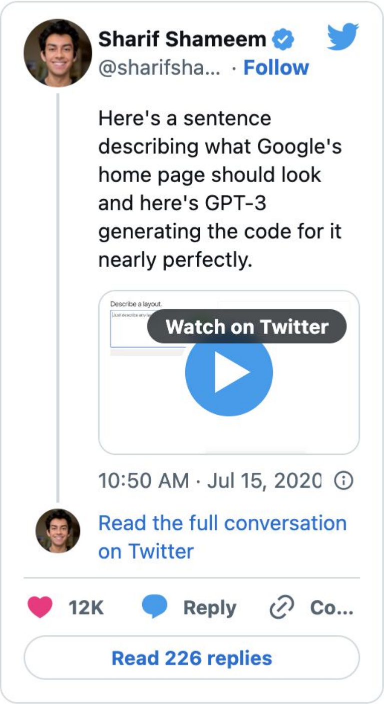
In-context Learning

This idea of prompting with demonstrations was called **In-context learning** in the original GPT-3 paper.

On many NLP benchmarks, in-context learning is **competitive with models trained with much more labeled data** and is SOTA on LAMBADA (commonsense sentence completion) and TriviaQA.

It enabled **various applications**:

Writing code from natural language descriptions

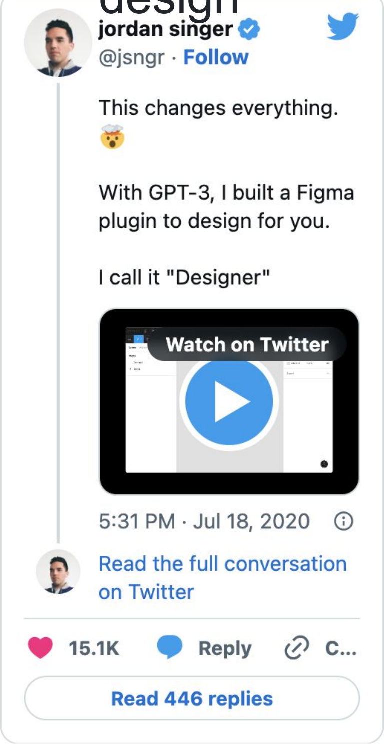


Sharif Shameem
@sharifsha...

Here's a sentence describing what Google's home page should look and here's GPT-3 generating the code for it nearly perfectly.

10:50 AM · Jul 15, 2020

App design



jordan singer
@jsngr

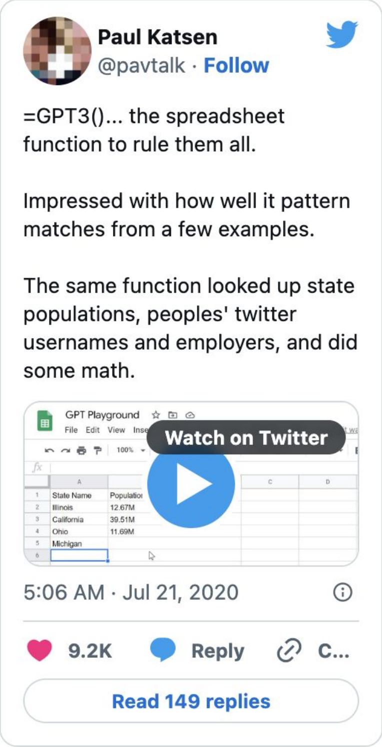
This changes everything. 🤯

With GPT-3, I built a Figma plugin to design for you.

I call it "Designer"

5:31 PM · Jul 18, 2020

Generalizing spreadsheet functions



Paul Katsen
@pavtalk

=GPT3()... the spreadsheet function to rule them all.

Impressed with how well it pattern matches from a few examples.

The same function looked up state populations, peoples' twitter usernames and employers, and did some math.

5:06 AM · Jul 21, 2020

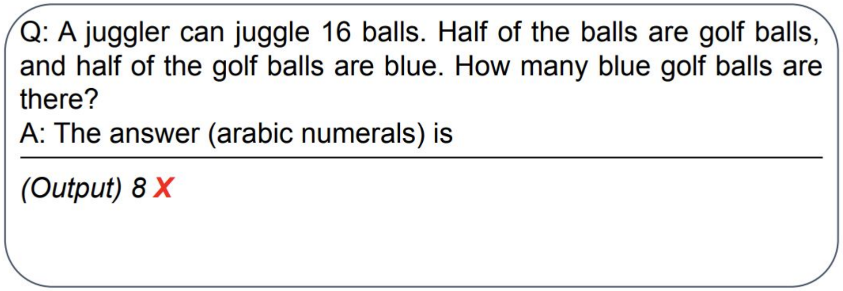
Chain of Thought Prompting

An approach to solve **multi-step reasoning** problems via prompting.

Key idea: Eliciting the model to produce a step-by-step solution of a problem can lead to a more accurate final answer (Wei et al. 2022)

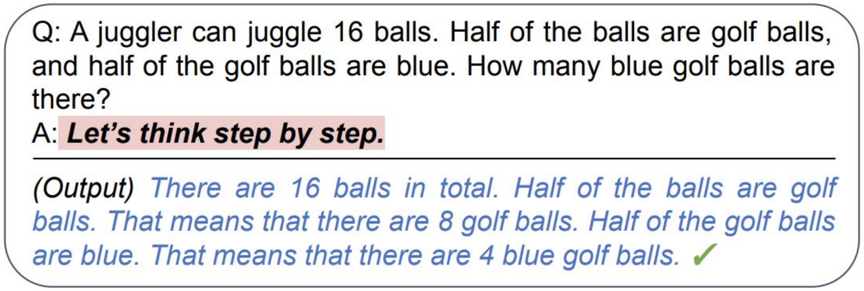
So, if we prompt the model to reason step-by step, it might do well at multi-step reasoning problems.

- **A solution:** Simply by adding “Let’s think step-by-step” to the prompt before the model’s answer



Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

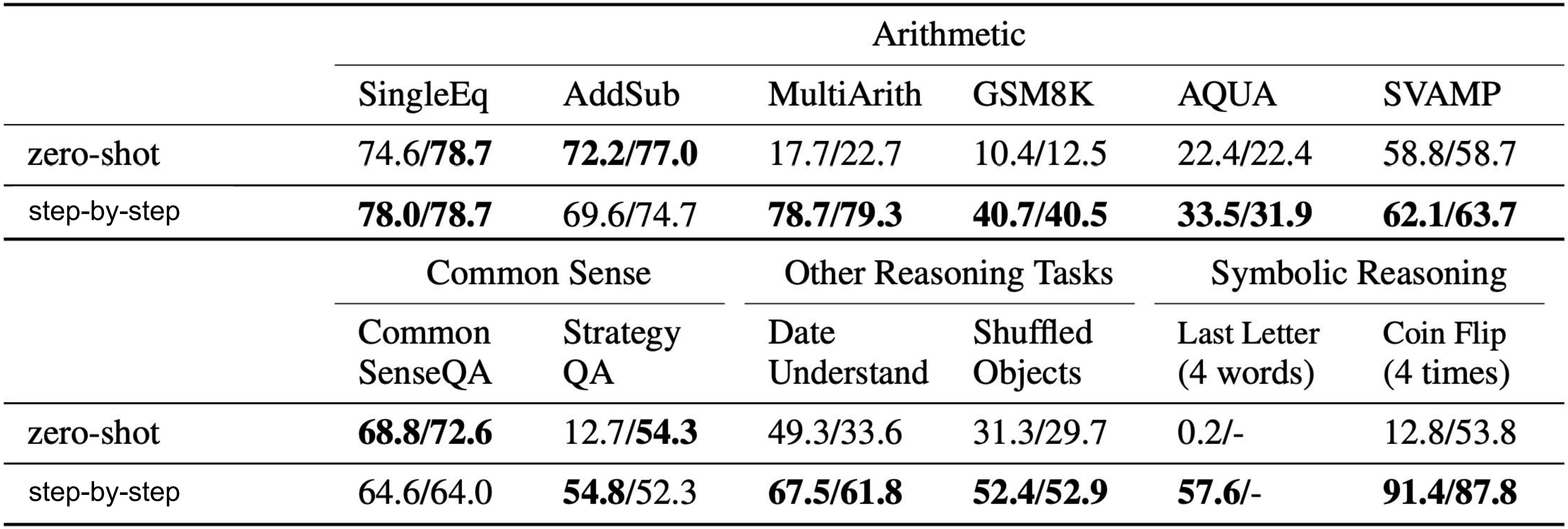
(Output) *8* ✗



Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let’s think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Thinking step-by-step works

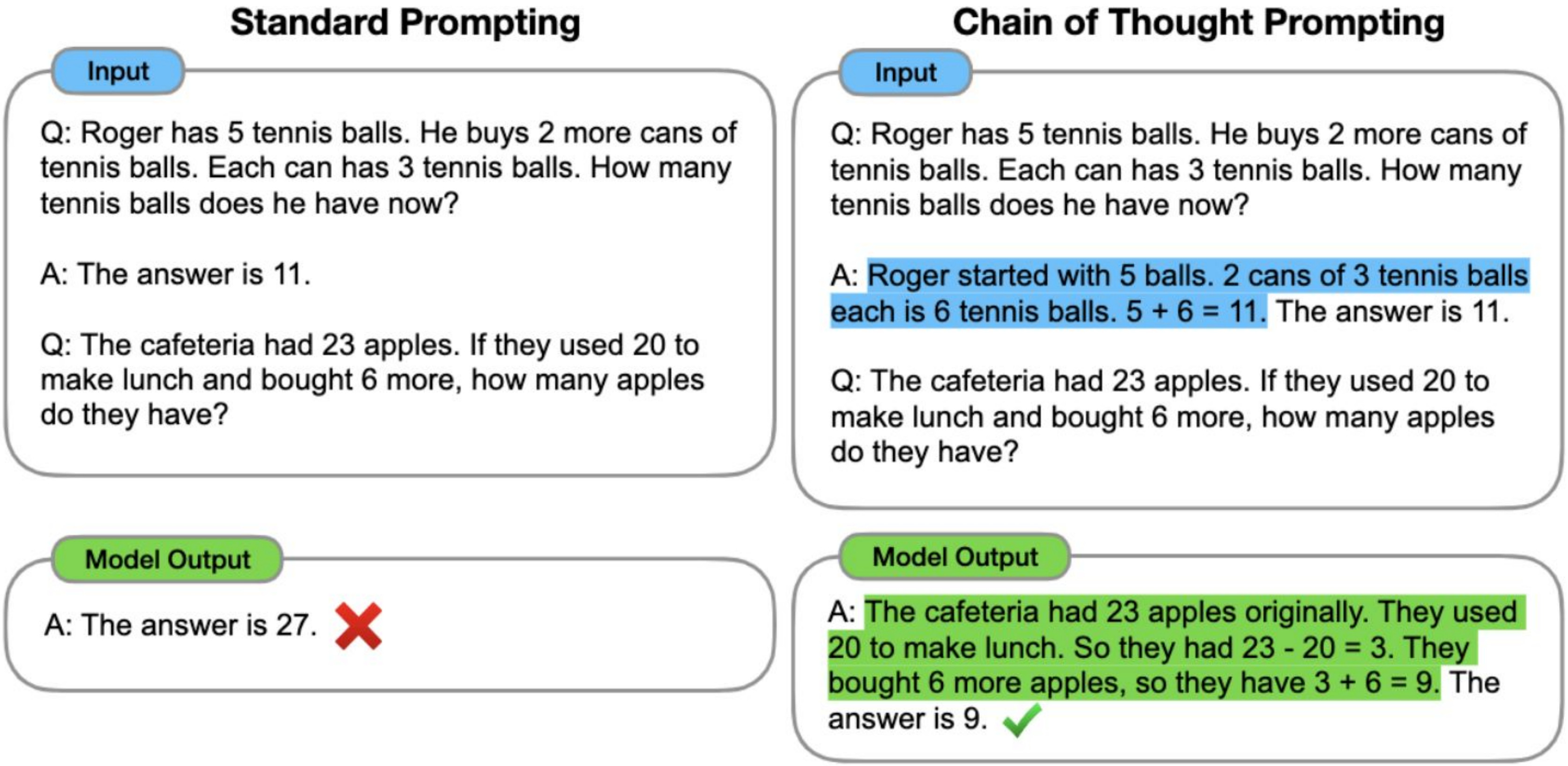


	Arithmetic					
	SingleEq	AddSub	MultiArith	GSM8K	AQUA	SVAMP
zero-shot	74.6/**78.7**	**72.2/77.0**	17.7/22.7	10.4/12.5	22.4/22.4	58.8/58.7
step-by-step	**78.0/78.7**	69.6/74.7	**78.7/79.3**	**40.7/40.5**	**33.5/31.9**	**62.1/63.7**

	Common Sense		Other Reasoning Tasks		Symbolic Reasoning	
	Common SenseQA	Strategy QA	Date Understand	Shuffled Objects	Last Letter (4 words)	Coin Flip (4 times)
zero-shot	**68.8/72.6**	12.7/**54.3**	49.3/33.6	31.3/29.7	0.2/-	12.8/53.8
step-by-step	64.6/64.0	**54.8**/52.3	**67.5/61.8**	**52.4/52.9**	**57.6/-**	**91.4/87.8**

Chain of Thought Prompting

This idea can be combined with the idea of including demonstrations in the prompt (Wei et al. 2022). This is called **Chain of Thought (CoT) prompting**.



Standard Prompting

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

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

Model Output

A: The answer is 27. ✗

Chain of Thought Prompting

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

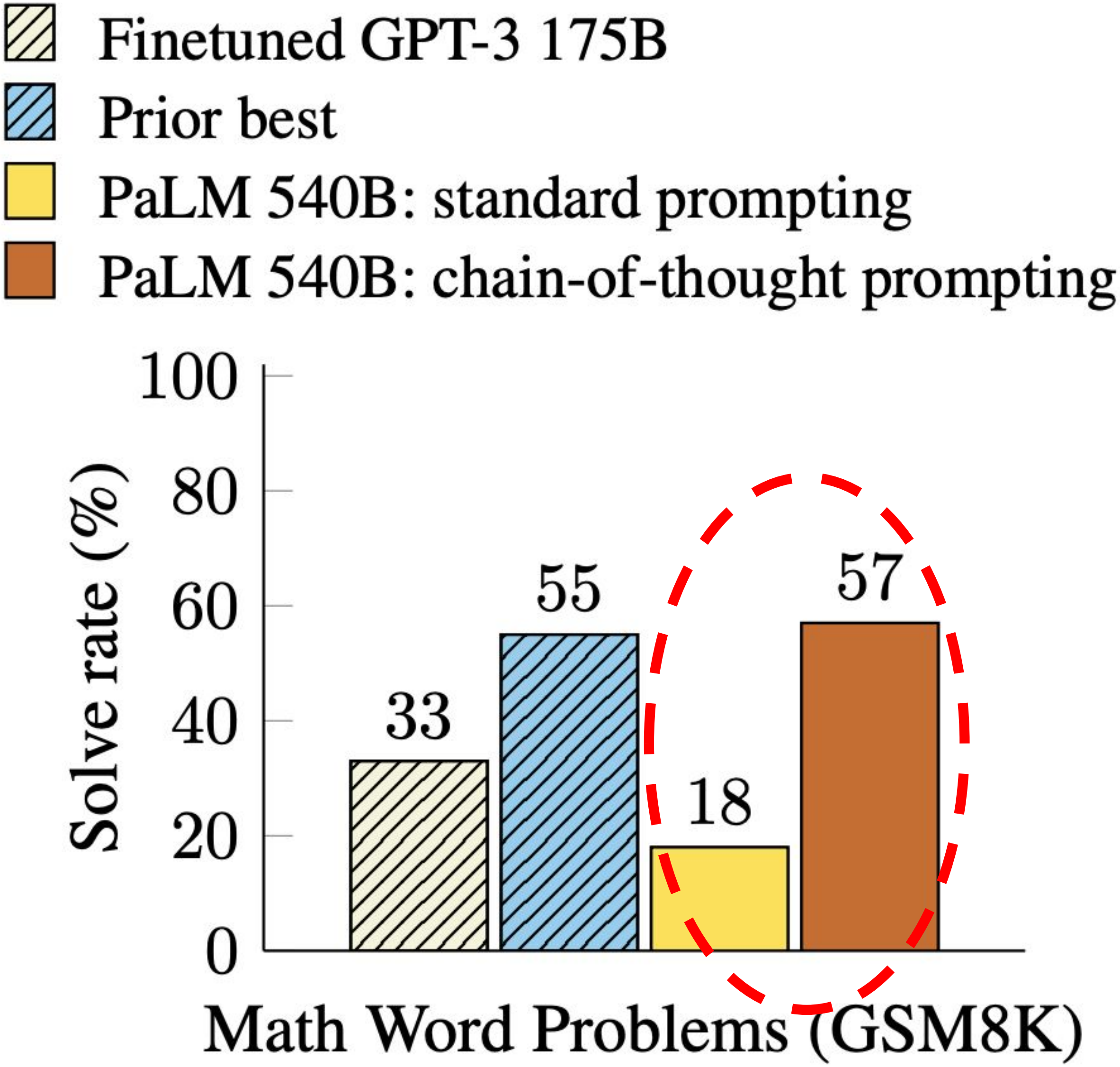
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

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

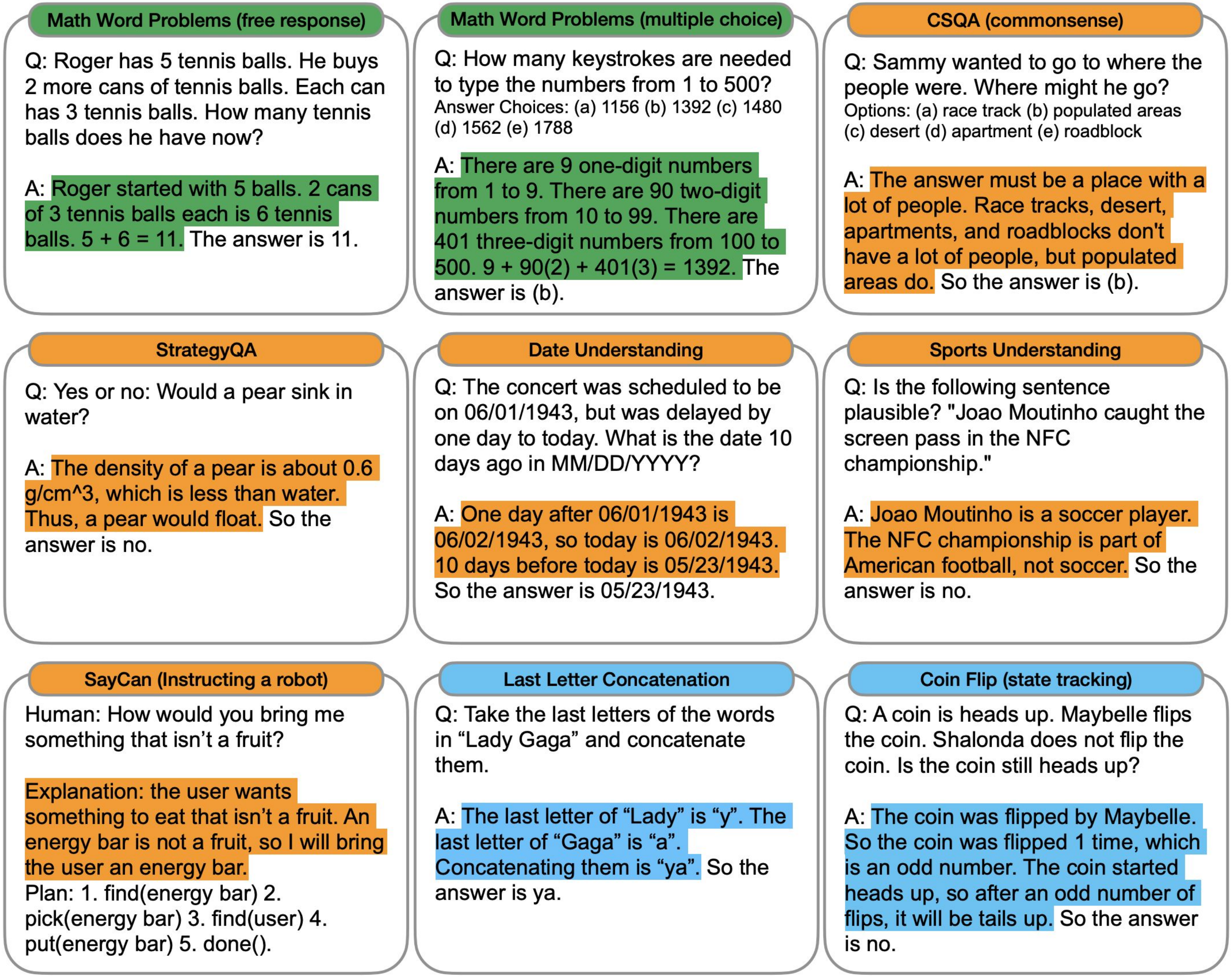
Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Chain of Thought Prompting



Chain of Thought Prompting



Math Word Problems (free response)

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Math Word Problems (multiple choice)

Q: How many keystrokes are needed to type the numbers from 1 to 500?
Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. 9 + 90(2) + 401(3) = 1392. The answer is (b).

CSQA (commonsense)

Q: Sammy wanted to go to where the people were. Where might he go?
Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock

A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).

StrategyQA

Q: Yes or no: Would a pear sink in water?

A: The density of a pear is about 0.6 g/cm^3, which is less than water. Thus, a pear would float. So the answer is no.

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

Sports Understanding

Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."

A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.

SayCan (Instructing a robot)

Human: How would you bring me something that isn't a fruit?

Explanation: the user wants something to eat that isn't a fruit. An energy bar is not a fruit, so I will bring the user an energy bar.
Plan: 1. find(energy bar) 2. pick(energy bar) 3. find(user) 4. put(energy bar) 5. done().

Last Letter Concatenation

Q: Take the last letters of the words in "Lady Gaga" and concatenate them.

A: The last letter of "Lady" is "y". The last letter of "Gaga" is "a". Concatenating them is "ya". So the answer is ya.

Coin Flip (state tracking)

Q: A coin is heads up. Maybelle flips the coin. Shalonda does not flip the coin. Is the coin still heads up?

A: The coin was flipped by Maybelle. So the coin was flipped 1 time, which is an odd number. The coin started heads up, so after an odd number of flips, it will be tails up. So the answer is no.

Thanks!