



CS 839: Foundation Models **Prompting I**

Fred Sala

University of Wisconsin-Madison

Sept. 26, 2024

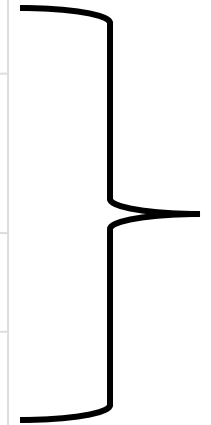
Announcements

- **Logistics:**

- Homework 1 is ongoing!
- [Llama 3.2](#) out ([try it!](#))

- **Class roadmap:**

Thursday Sept. 26	Prompting I
Tuesday Oct. 1	Prompting II
Thursday Oct. 3	Specialization
Tuesday Oct. 8	Alignment
Thursday Oct. 10	Efficient Training



Language & Foundation Models

Outline

- **Intro to Prompting-Review from Last Time**

- Terminology: zero-shot, few-shot, in-context, etc, prompt characteristics: format, examples, orders

- **Improving and Extending Prompting**

- Searching for good prompts, techniques for continuous/soft prompts, ensembling

- **Intro to Chain-of-Thought**

- Basic idea, zero-shot and few-shot, choosing examples for few-shot, tree-of-thoughts

Outline

- **Intro to Prompting-Review from Last Time**

- Terminology: zero-shot, few-shot, in-context, etc, prompt characteristics: format, examples, orders

- **Improving and Extending Prompting**

- Searching for good prompts, techniques for continuous/soft prompts, ensembling

- **Intro to Chain-of-Thought**

- Basic idea, zero-shot and few-shot, choosing examples for few-shot, tree-of-thoughts

Prompting: Ask Your Model


Essentially, ask your model to perform your goal task

Example: sentiment analysis task

- Prompt: “Text: The visuals were lacking and the characters felt flat. Sentiment:”

- Result: “Negative”

Default (GPT-3.5)

 Text: The visuals were lacking and the characters felt flat. Sentiment:

 Negative

Prompting: Zero-shot vs Few-shot

Terminology:

- **Zero-shot:** No “examples” provided to the model.
- **Few-shot/in-context learning:** Provide “examples”

Input: Subpar acting. Sentiment: Negative

Input: Beautiful film. Sentiment: Positive

Input: Amazing. Sentiment:

Zhao et al '21



Positive

Prompting: Few-shot vs. In-context learning

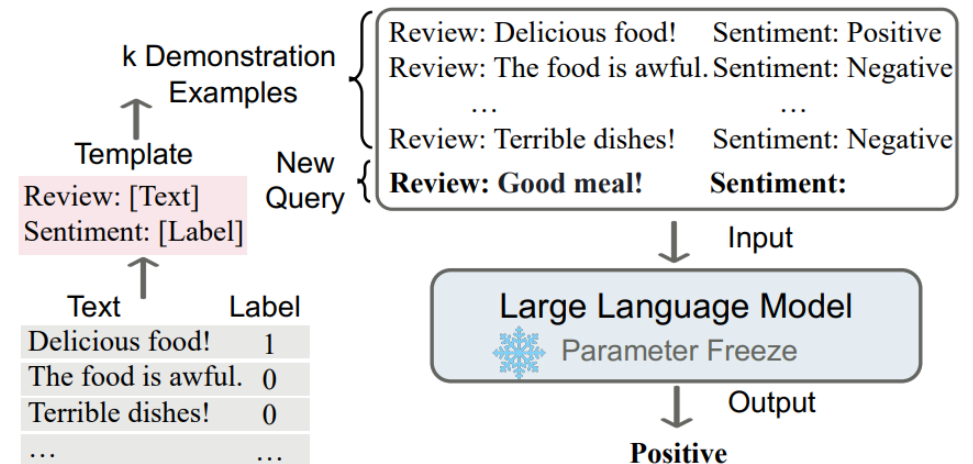
Terminology conflicts! Note: we have a set of labeled examples. Could **fine-tune!**

Few-shot: *sometimes* means fine-tune on this dataset, then prompt

In-context learning: do not fine-tune. Model weights unchanged.

```
Text: (lawrence bounces) all over the stage, dancing,  
Sentiment: positive  
  
Text: despite all evidence to the contrary, this clun  
Sentiment: negative  
  
Text: for the first time in years, de niro digs deep  
Sentiment: positive
```

Weng / SST

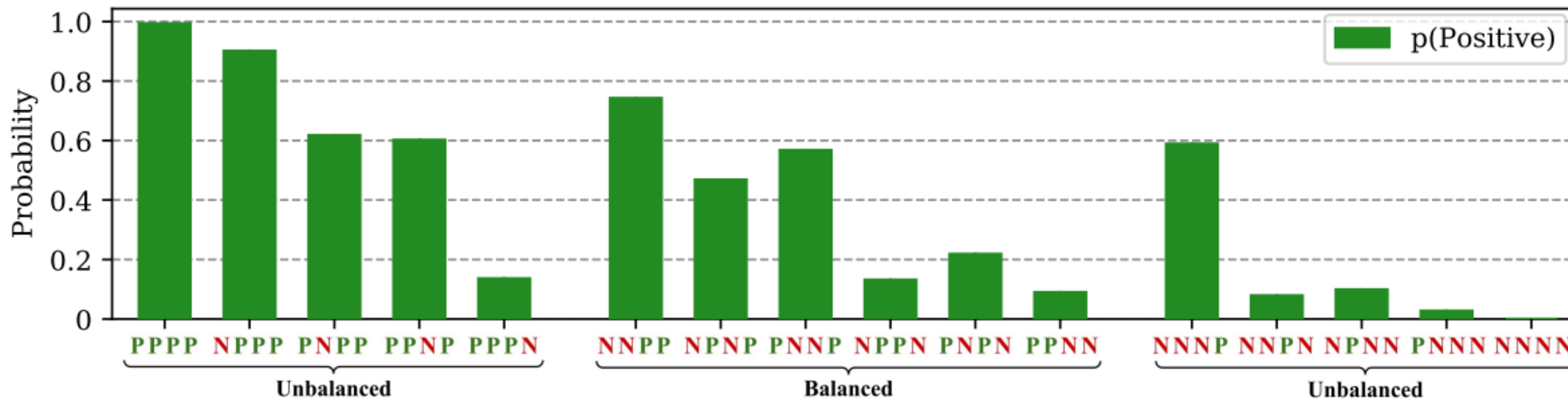


Dong et al, '23

Few-Shot Choices

Examples/structure affect performance:

1. Prompt **format** (affects everything)
2. **Choice** of examples
3. **Order** of examples (permutation)



1. Prompt Formats

The choice of model affects the prompt format

Masked language model: “Cloze”-style prompt

- “I love this movie, it is a [Z] movie:”

Left-to-right language model: prefix prompt

- “I love this movie. What is the sentiment of this review?”

Note: eval datasets have pre-created prompts.

- LAMA (LAnguage Model Analysis): Cloze prompts



1. Prompt Formats: Recent Models

Modern instruction-tuned models have more complex instructions/formats

- **The good:** more natural way to tell the model what to do
- **The bad:** searching over formats/templates increasingly challenging
 - *Example: (White et al, '23): "From now on, I would like you to ask me questions to deploy a Python application to AWS. When you have enough information to deploy the application, create a Python script to automate the deployment."*

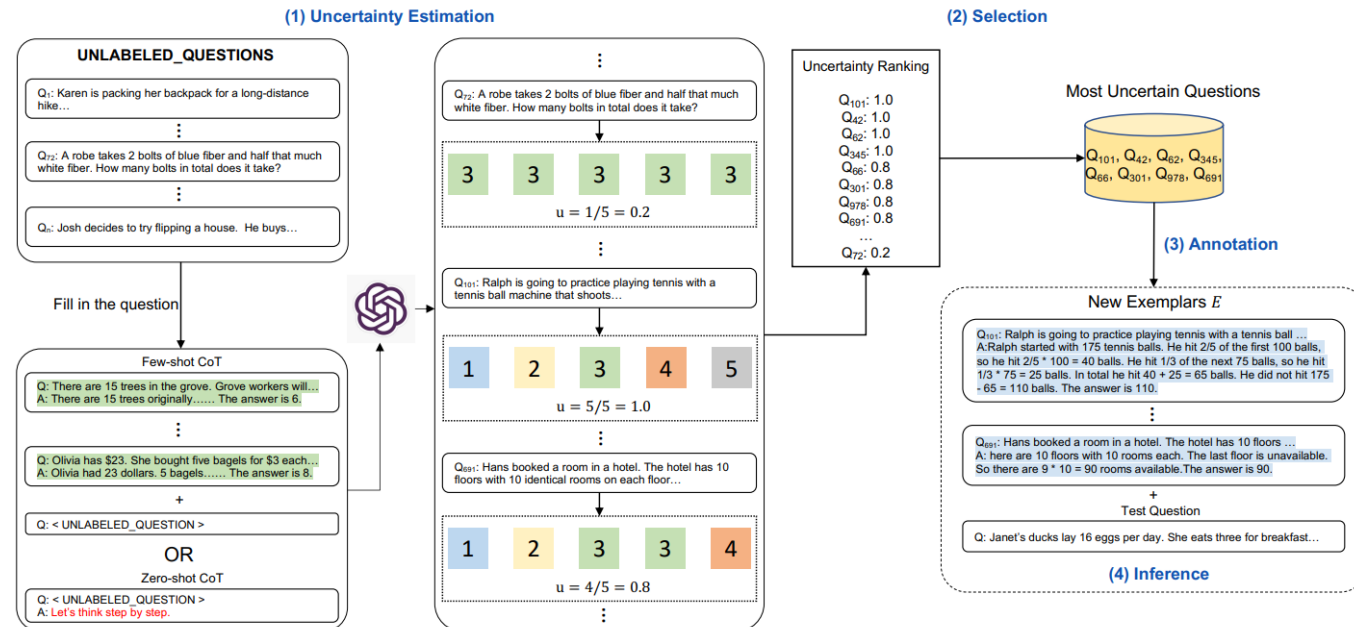
2. Choice of Examples

How to pick appropriate examples in few-shot?

- **Note:** only a “small” number of examples can be shown, unlike in supervised learning.

Many options. Sampling:

- Liu et al, '21: kNN in embedding space (semantic similarity)
- Su et al, '22: Encourage diversity in embeddings
- Diao et al, '23: “Active prompting”



Diao et al '23

3. Order of Examples

What order to show them to the model?

**Fantastically Ordered Prompts and Where to Find Them:
Overcoming Few-Shot Prompt Order Sensitivity**

Yao Lu[†] Max Bartolo[†] Alastair Moore[‡] Sebastian Riedel[†] Pontus Stenetorp[†]

[†]University College London [‡]Mishcon de Reya LLP

{yao.lu,m.bartolo,s.riedel,p.stenetorp}@cs.ucl.ac.uk

alastair.moore@mishcon.com

• Findings:

- Model size doesn't guarantee low-variance
- Adding more examples doesn't reduce variance
- Good prompts don't transfer from one model to another 😞
- Good orders don't transfer



Break & Questions

Outline

- **Intro to Prompting-Review from Last Time**

- Terminology: zero-shot, few-shot, in-context, etc, prompt characteristics: format, examples, orders

- **Improving and Extending Prompting**

- Searching for good prompts, techniques for continuous/soft prompts, ensembling

- **Intro to Chain-of-Thought**

- Basic idea, zero-shot and few-shot, choosing examples for few-shot, tree-of-thoughts

Hard Prompting

Also called **zero-shot**.

- Note: terminology conflict with another area called zero-shot learning

“Hard prompt discovery is a specialized alchemy, with many good prompts being discovered by trial and error, or sheer intuition

(Wen et al '23)

- Note: not just for language models!



Optimize Prompt ↘

 cuddly teddy skateboarding
comforting nyc led cl

↑ Generate Image

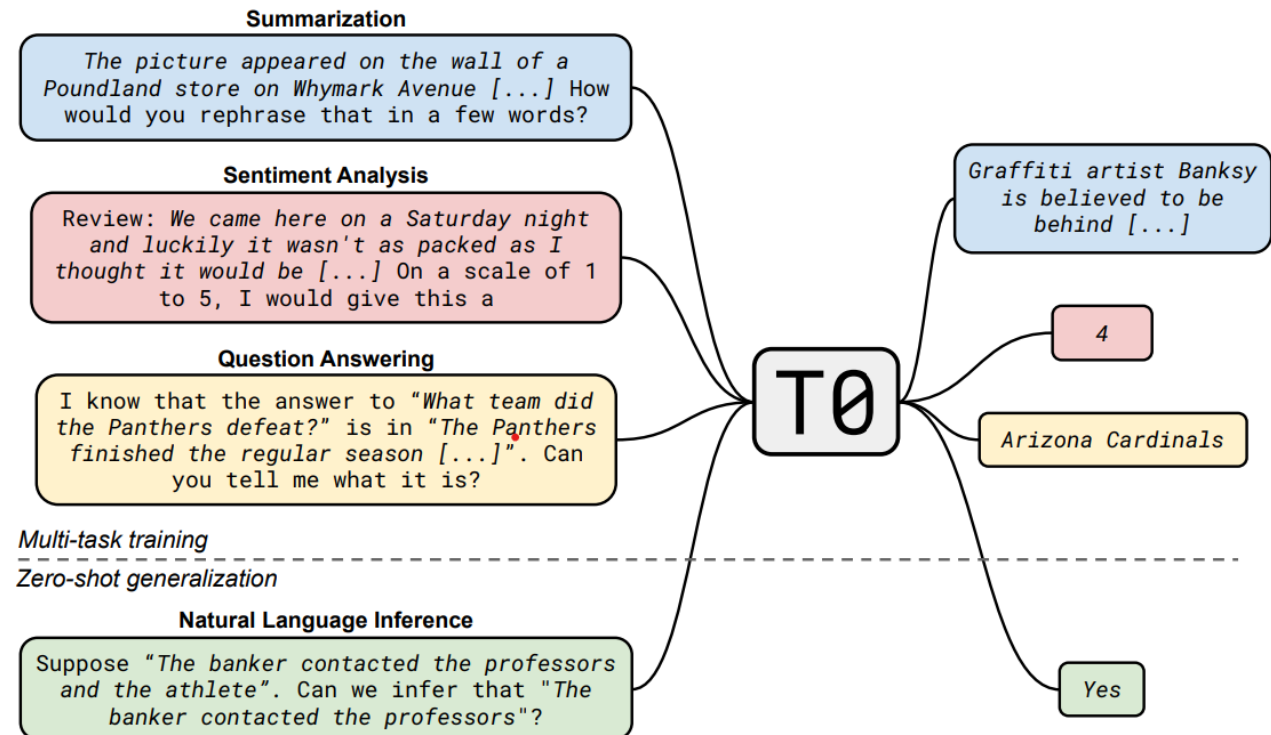
Zero-shot Generalization

Most exciting aspect of zero-shot: don't need to have been explicitly trained or fine-tuned.

• Example: Multitask Prompted Training Enables Zero-Shot Task Generalization

Recipe

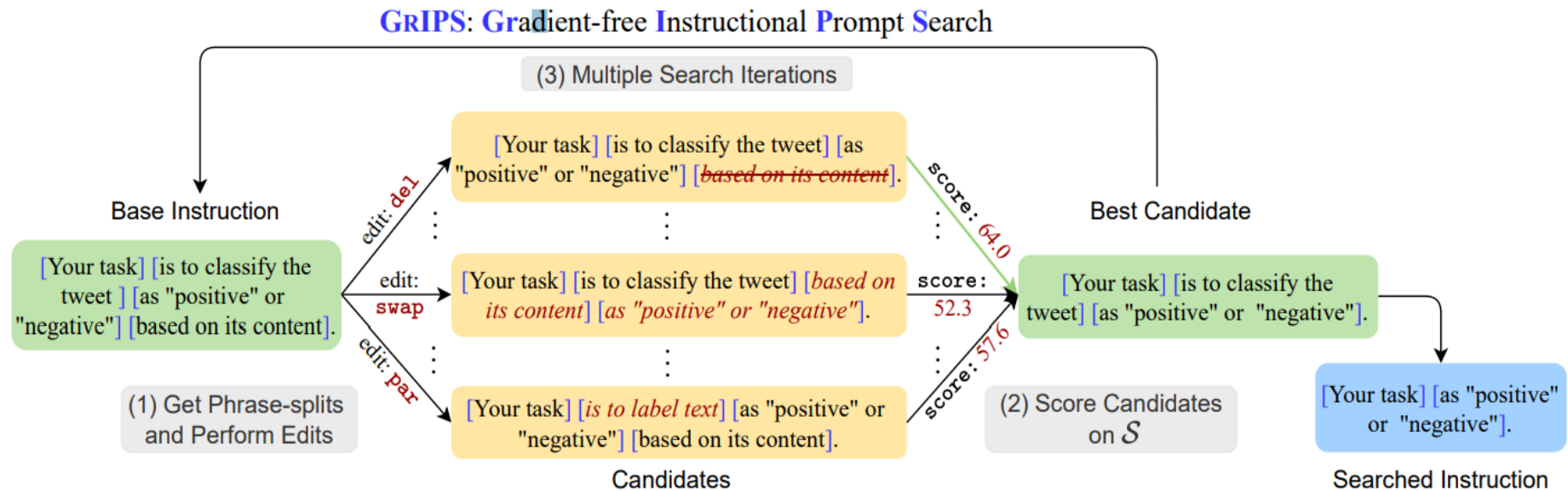
- Pretrain
- Fine-tune
 - Multitask



Hard Prompting: Discrete Optimization

Sometimes, can avoid gradients

- Random search
- Greedy



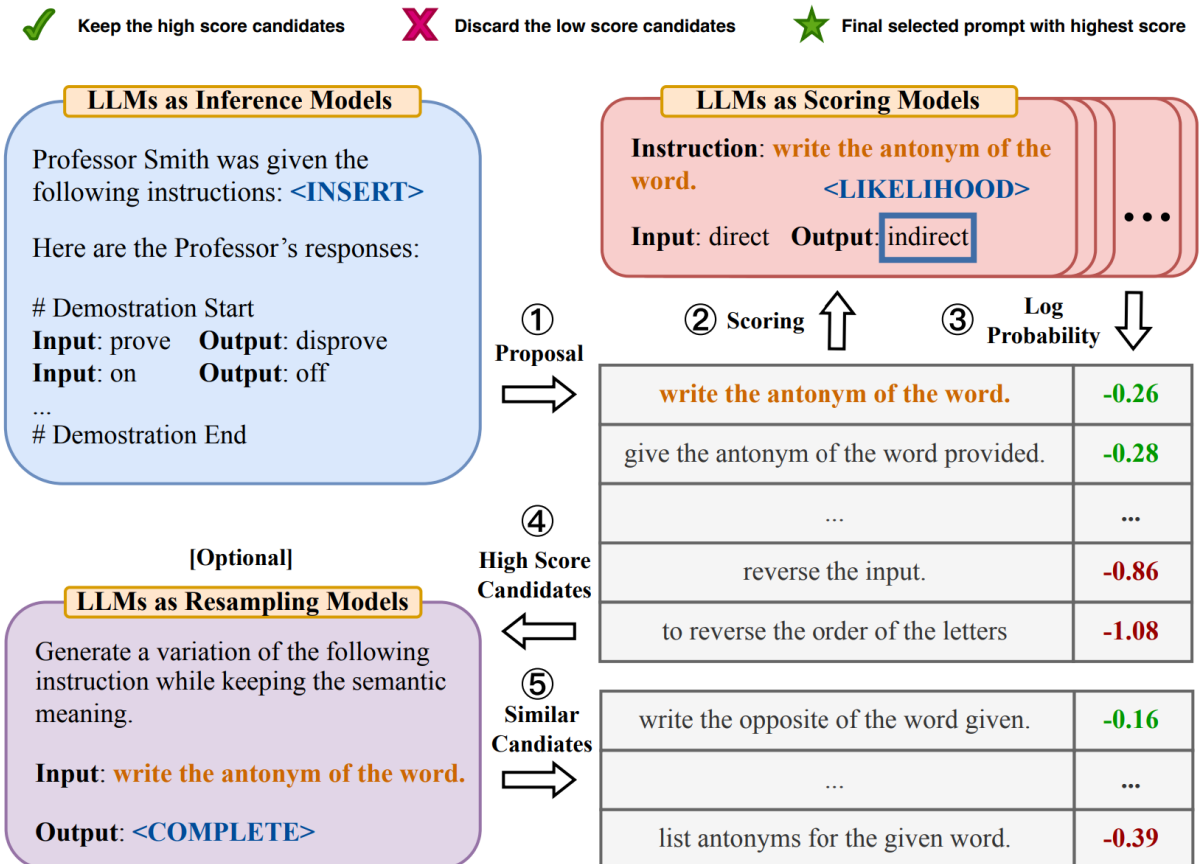
More on Auto-Prompting

LLMs as “prompt engineers” (Zhou et al, ‘23)

- Use an LLM to generate candidate instructions (prompts)
- Evaluate them externally
- Select best candidate.
- Optionally iterate.

Example Output:

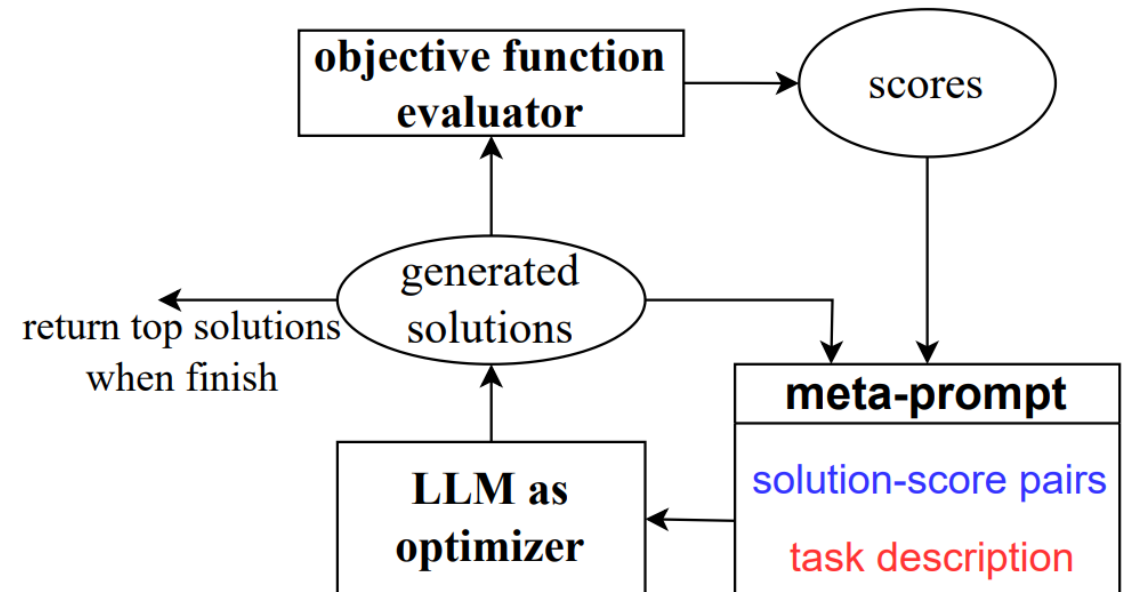
- “Let’s work this out in a step by step way to be sure we have the right answer.”



More on Auto-Prompting

LLMs as “optimizers” (Yang et al, ‘23)

- Use the LLM to guess solutions to an optimization problem
 - Evaluate them externally
 - Run in a loop with few-shot
-
- Can use for
 - Standard optimization problems
 - Tougher settings like prompts



LLMs as Optimizers: Example

Example:

I have some texts along with their corresponding scores. The texts are arranged in ascending order based on their scores, where higher scores indicate better quality.

text:
Let's figure it out!
score:
61

text:
Let's solve the problem.
score:
63

(... more instructions and scores ...)

The following exemplars show how to apply your text: you replace <INS> in each input with your text, then read the input and give an output. We say your output is wrong if your output is different from the given output, and we say your output is correct if they are the same.

input:
Q: Alannah, Beatrix, and Queen are preparing for the new school year and have been given books by their parents. Alannah has 20 more books than Beatrix. Queen has $\frac{1}{5}$ times more books than Alannah. If Beatrix has 30 books, how many books do the three have together?
A: <INS>
output:
140

(... more exemplars ...)

Write your new text that is different from the old ones and has a score as high as possible. Write the text in square brackets.



Meta-instructions



Trajectory points



Problem to be solved

LLMs as Optimizers: Prompt Optimization

Resulting trajectory

- “Solve the following problems using the given information.” at Step 2 with training accuracy 59.8;
- “Solve the following problems by applying the given information and using the appropriate mathematical operations.” at Step 3 with training accuracy 64.0;
- “Let’s read the problem carefully and identify the given information. Then, we can create an equation and solve for the unknown variable.” at Step 4 with training accuracy 67.0;
- “I’m always down for solving a math word problem together. Just give me a moment to read and understand the problem. Then, I’ll create an equation that models the problem, which I’ll solve for the unknown variable. I also may or may not use some helpful diagrams or visuals to understand the problem. Lastly, be sure to allow me some time to carefully check my work before submitting any responses!” at Step 29 with training accuracy 70.1.

Ours

PaLM 2-L

PaLM
2-T-TT

A_begin

Take a deep breath and work on this problem step-by-step.

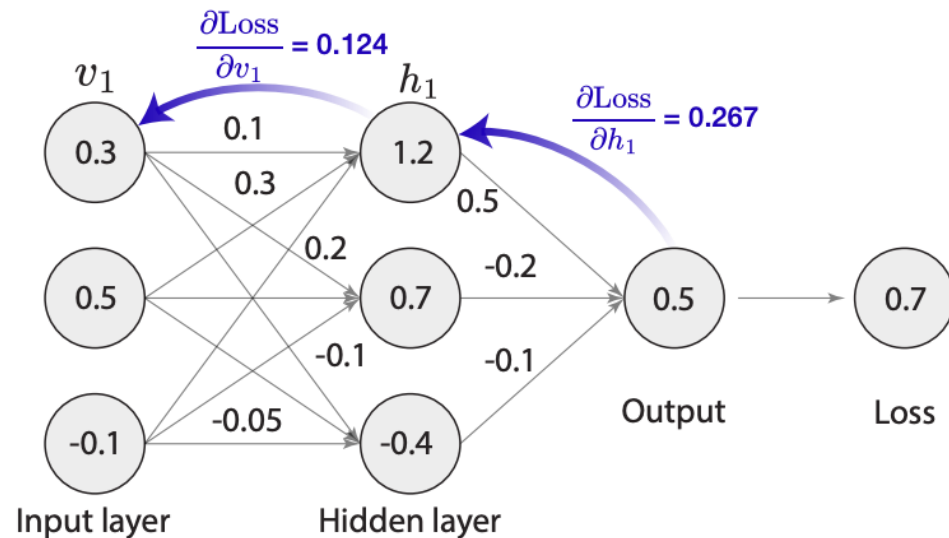
80.2

Hard Prompting: Text Optimization

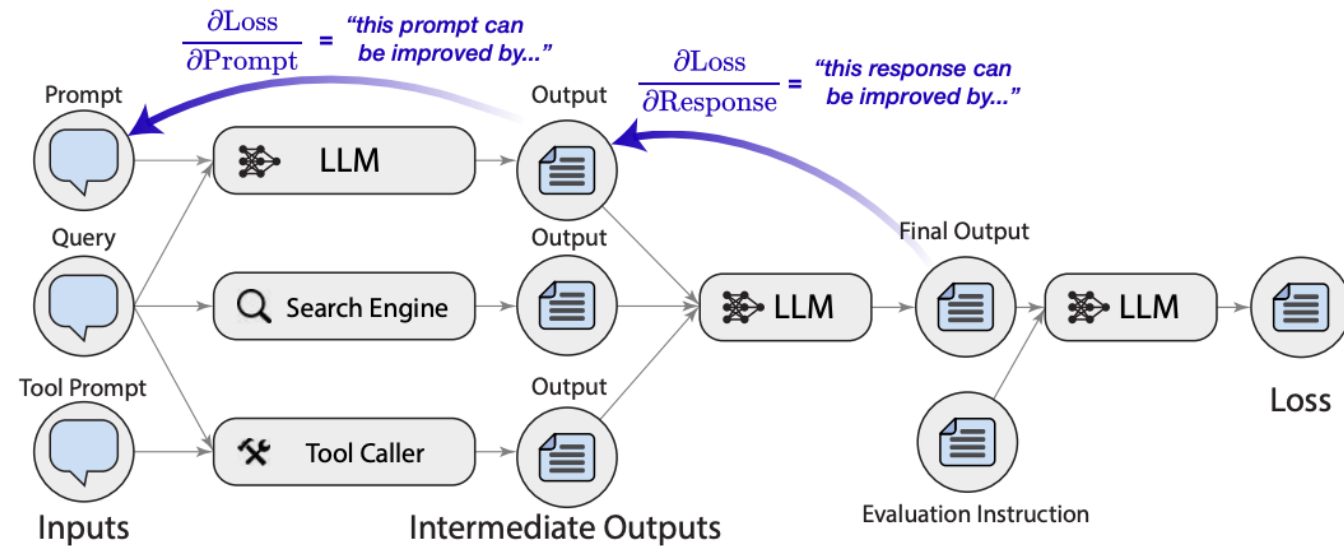
Sometimes, can even “generalize” autodiff’

- TextGrad

a Neural network and backpropagation using numerical gradients



b Blackbox AI systems and backpropagation using natural language ‘gradients’



Soft Prompting

Also called **continuous prompting**

Basic idea: insert some (non-language) parameters into prompt

- Train these parameters
- Do not directly correspond to words in prompt

Prefix-Tuning: Optimizing Continuous Prompts for Generation

Xiang Lisa Li
Stanford University
xlisali@stanford.edu

Percy Liang
Stanford University
плианг@cs.stanford.edu

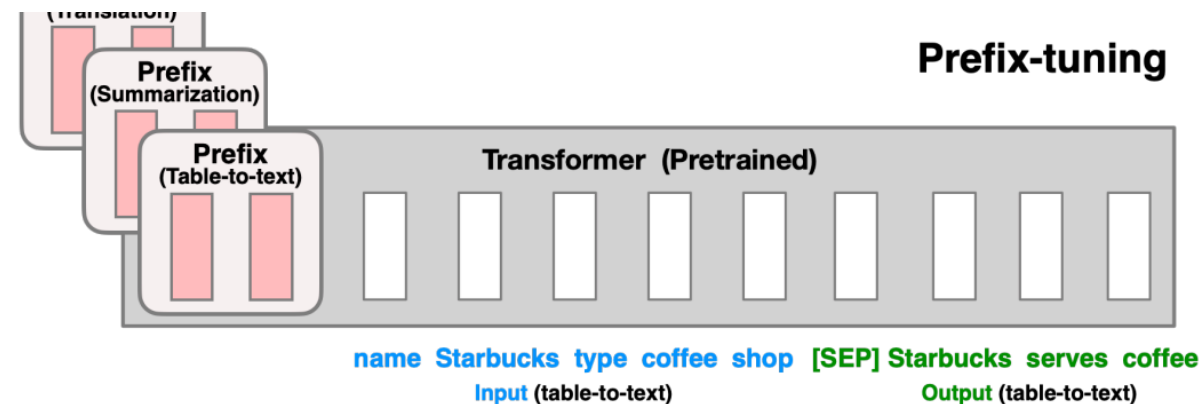
GPT Understands, Too

Xiao Liu^{*12} Yanan Zheng^{*12} Zhengxiao Du¹² Ming Ding¹² Yujie Qian³ Zhilin Yang⁴² Jie Tang¹²

Soft Prompting: Prefix-Tuning

Goal: create prefixes that *steer* models

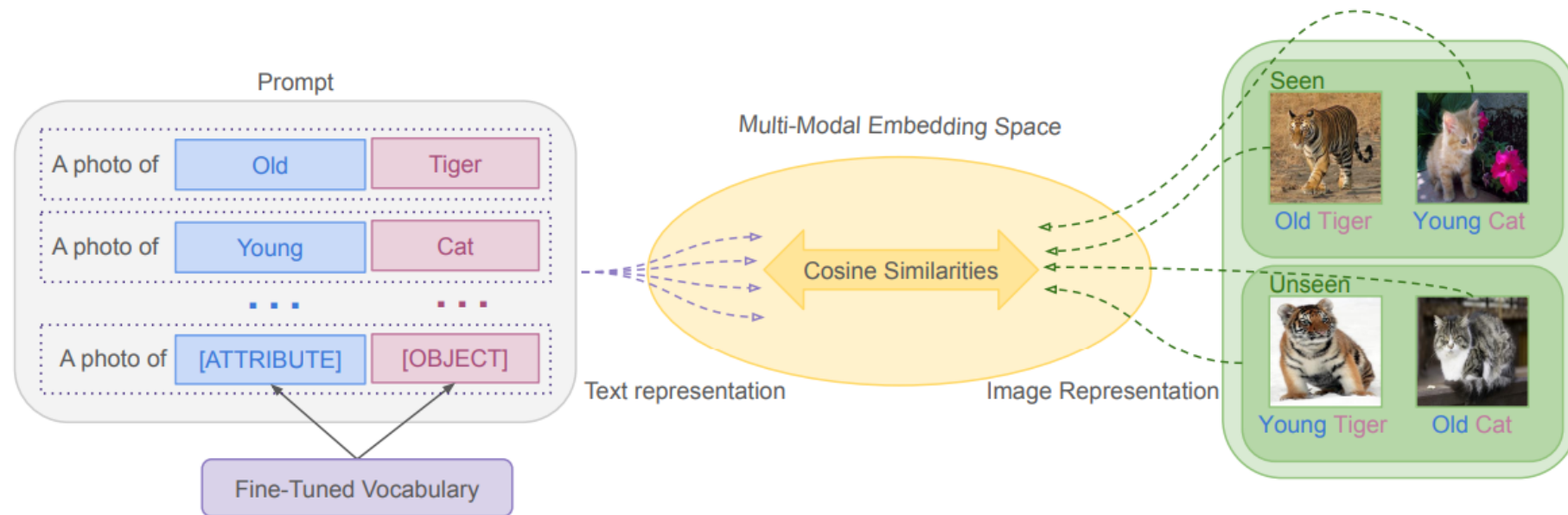
- Prefixes are trainable parameters
- Train one for each goal task, only store these new parameters
- Enables cheap adaptation of frozen language model



Soft Prompting: Composing

What about multimodal models?

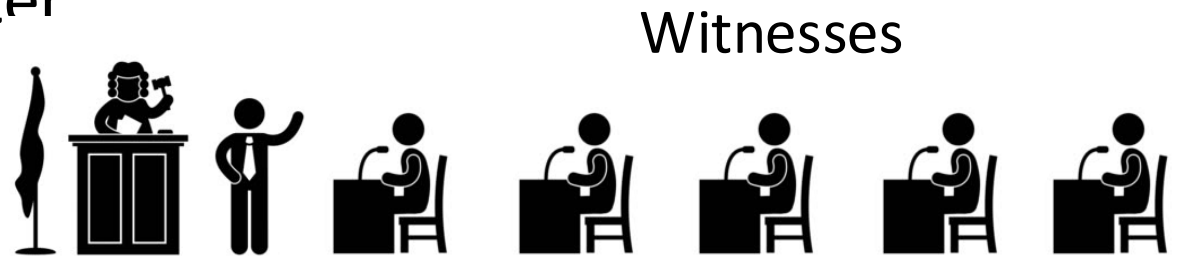
- Vision-language models like CLIP
- Not great composed concepts like *old tiger*
- Tune on [attribute] [object] pairs



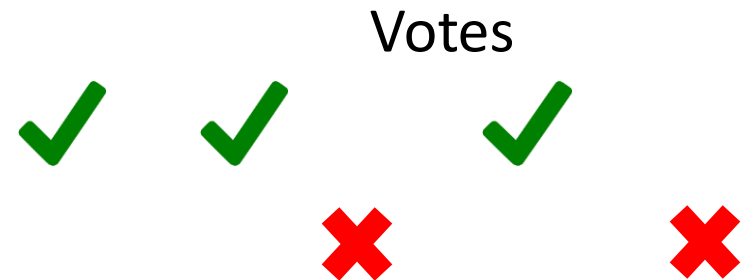
Ensembling Prompts

One prompt can give you an answer... but might be wrong

- One simple approach: get multiple samples
- From?
 - Change temperature parameter
 - Vary your prompts



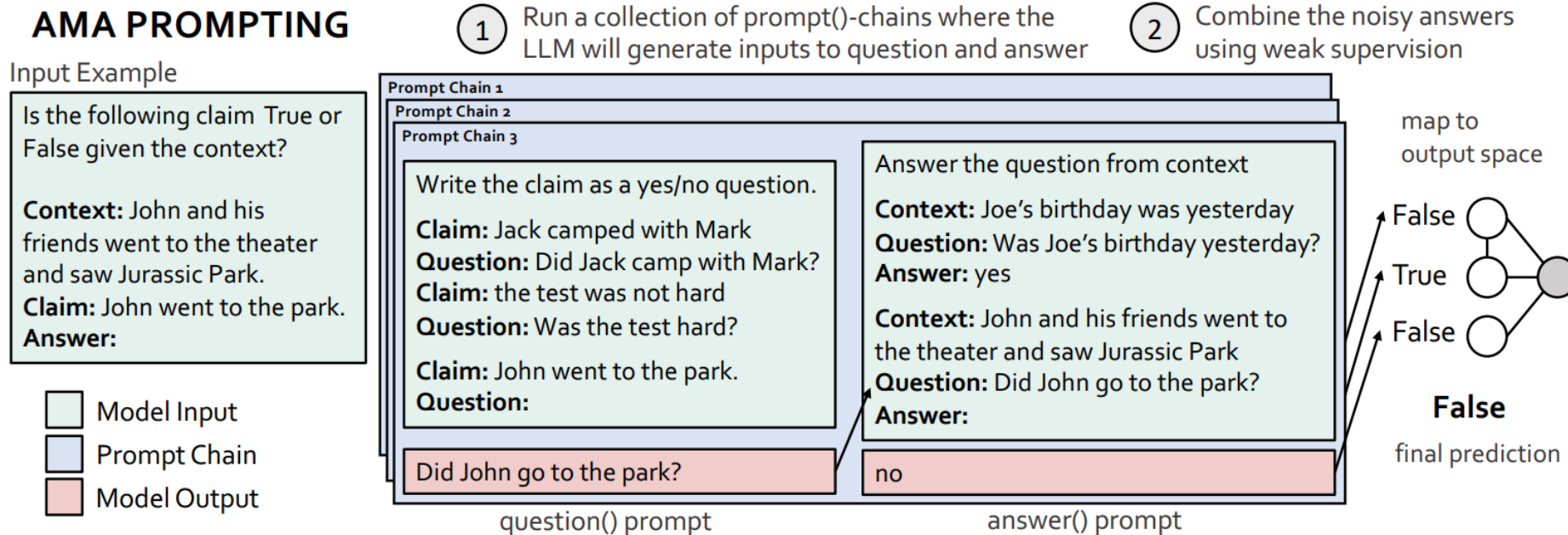
Then, run **majority vote**



Ensembling Prompts: Weighted Version

Downside of majority vote... most responses might be wrong

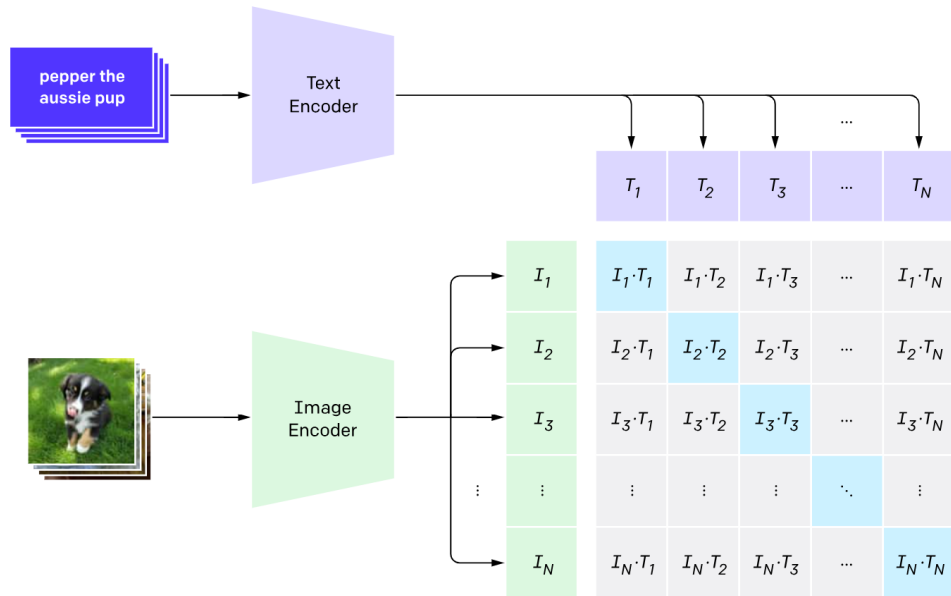
- Should weight them by how accurate they are



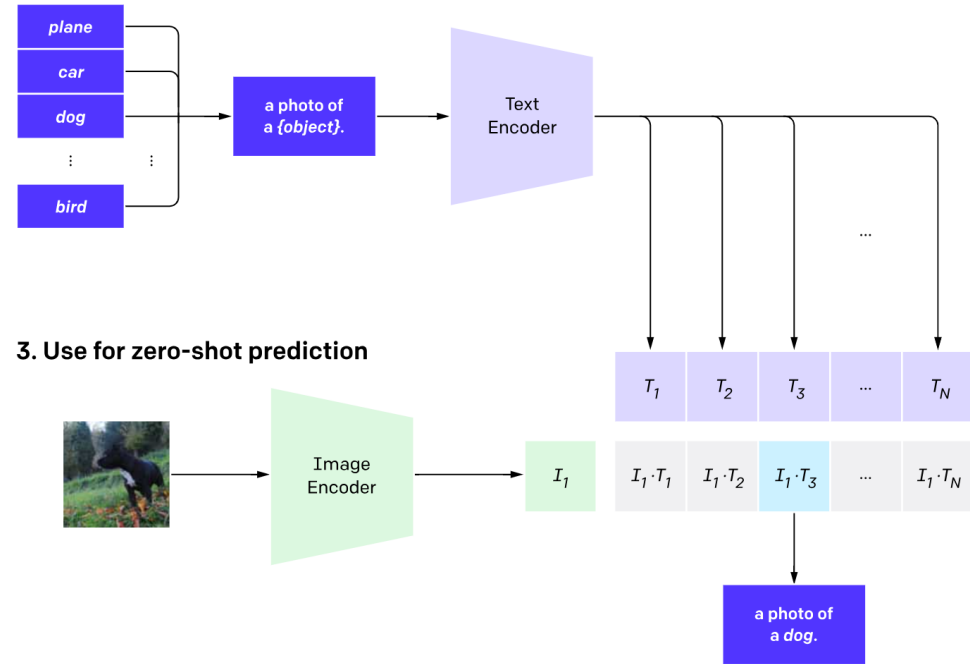
Prompting VLMs and Multimodal Models

Training and prediction in CLIP-style VLMs

1. Contrastive pre-training



2. Create dataset classifier from label text




3. Use for zero-shot prediction

How to Prompt VLMs?

Standard way: use pre-defined templates

- E.g., “a photo of a [X]”
- But, might struggle...

SUN397
television studio (90.2%) Ranked 1 out of 397 labels



- ✓ a photo of a **television studio**.
- ✗ a photo of a **podium indoor**.
- ✗ a photo of a **conference room**.
- ✗ a photo of a **lecture room**.
- ✗ a photo of a **control room**.

LLMs to Improve VLMs: Description

Static class descriptions may fail...

- Replace with descriptive features (Menon and Vondrick, '23)
 - Instead of “tiger”, include “stripes, claws, ...”

School bus

- a large, yellow vehicle
- the words "school bus" written on the side
- a stop sign that deploys from the side of the bus
- flashing lights on the top of the bus
- large windows

Shoe store

- a building with a sign that says "shoe store"
- a large selection of shoes in the window
- shoes on display racks inside the store
- a cash register
- a salesperson or customer

Volcano

- a large, cone-shaped mountain
- a crater at the top of the mountain
- lava or ash flowing from the crater
- a plume of smoke or ash rising from the crater

Barber shop

- a building with a large, open storefront
- a barber pole or sign outside the shop
- barber chairs inside the shop
- mirrors on the walls
- shelves or cabinets for storing supplies
- a cash register
- a waiting area for customers

Cheeseburger

- a burger patty
- cheese
- a bun
- lettuce
- tomato
- onion
- pickles
- ketchup
- mustard

Violin

- a stringed instrument
- typically has four strings
- a wooden body
- a neck and fingerboard
- tuning pegs
- a bridge
- a soundpost
- f-holes
- a bow

Pirate ship

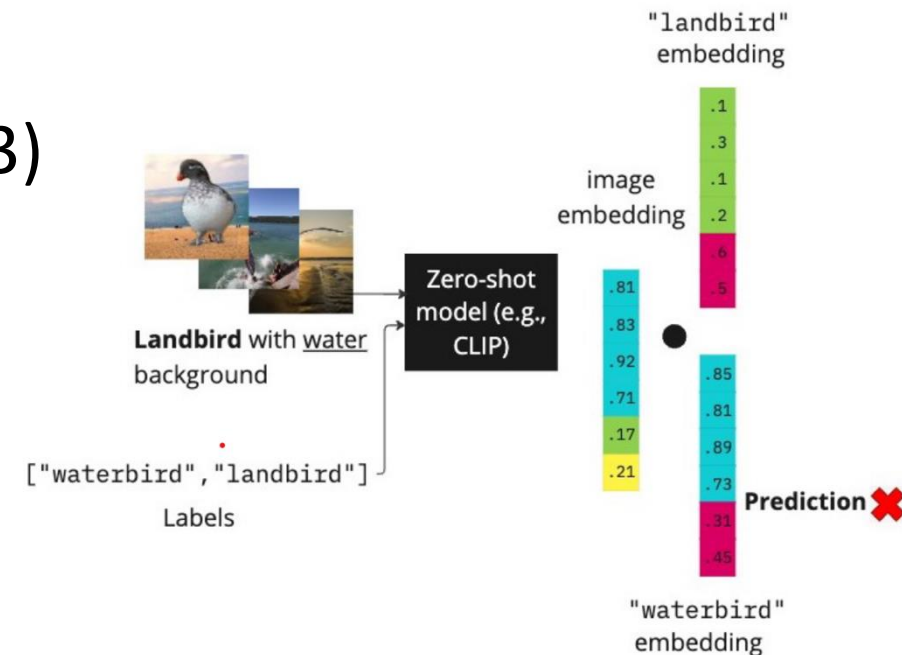
- a large, sailing vessel
- a flag with a skull and crossbones
- cannons on the deck
- a wooden hull
- portholes
- rigging
- a crow's nest

Figure 3: Examples of descriptor schema produced by GPT-3.

LLMs to Improve VLMs: Spurious Features

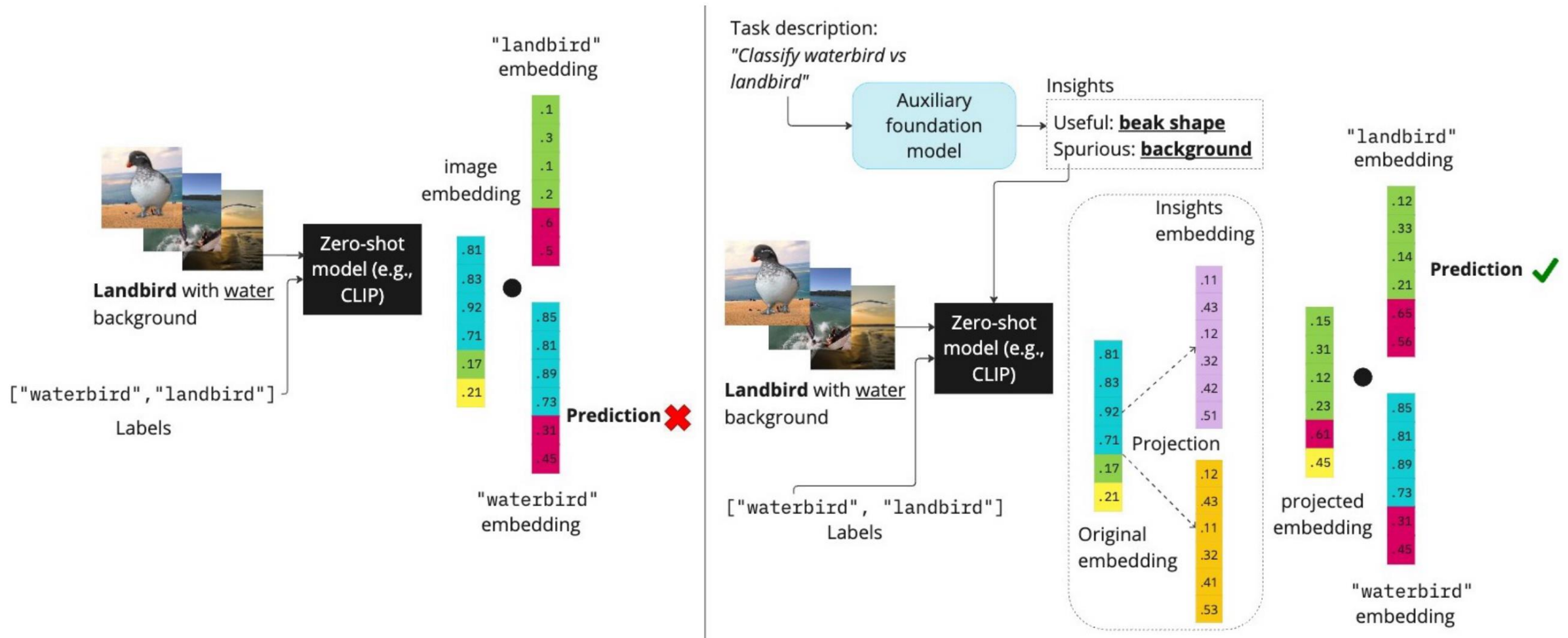
This helped with positives.

- What about **negatives** (i.e., spurious features?)
 - Example: waterbirds with CLIP
- Spurious correlations: generally a problem with all pretrained models
 - But LLMs can also tell us about this (Adila '23)



LLMs to Improve VLMs: Spurious Features

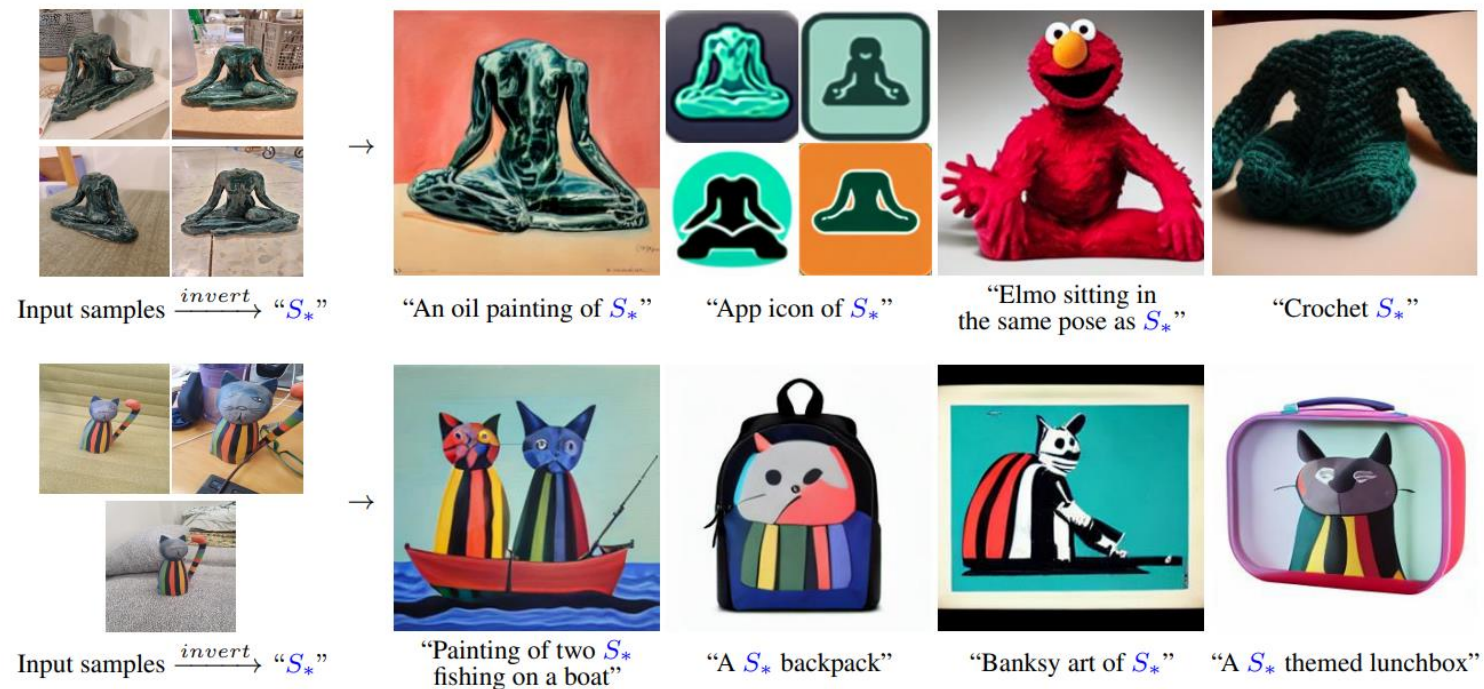
Modify embeddings used for prediction:



How to Prompt VLMs, Image Models?

Generally, all the methods for language cases apply

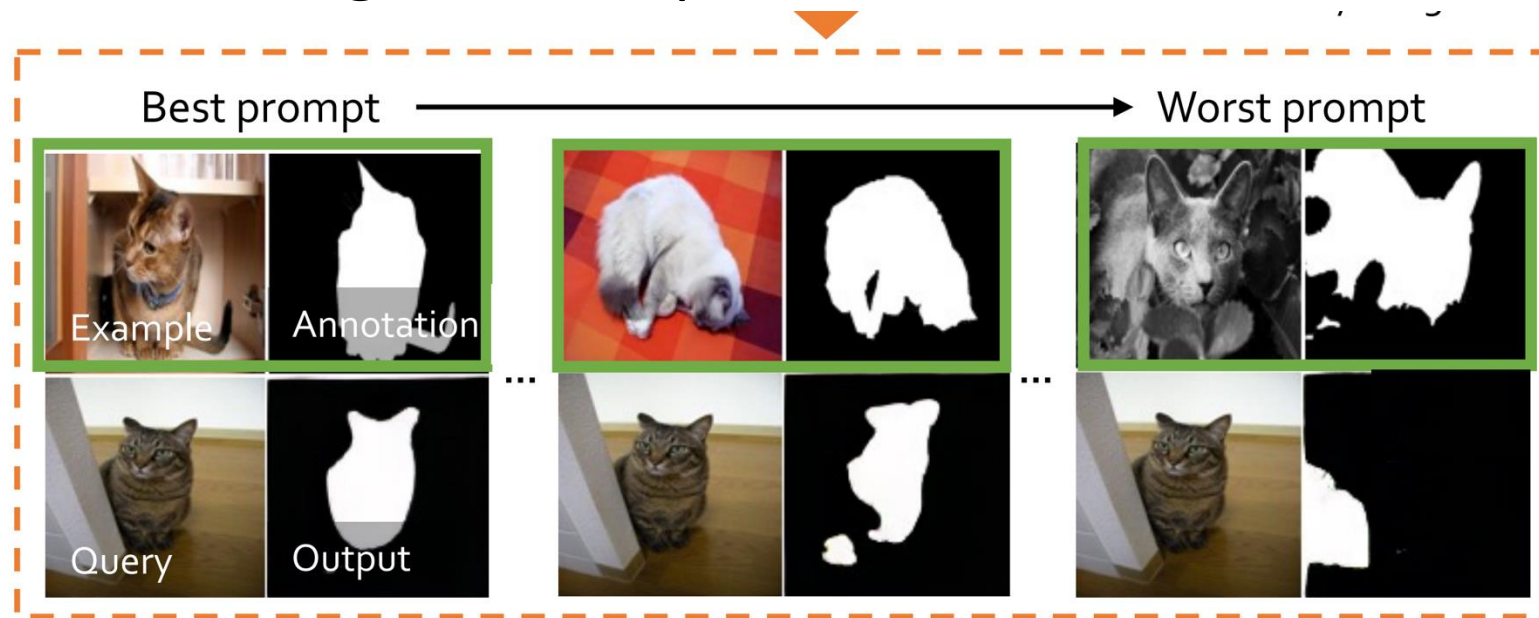
- Especially **soft prompts**
- Often part of “inversion” pipelines (Gal et al ‘22)



In-Context Learning for VLMs

Surprisingly in-context/few-shot also applies to visual models

- Standard questions apply here too:
 - How to select examples
 - What makes for a good example?



(a) Visual in-context learning is sensitive to prompt selection



Break & Questions

Outline

- **Intro to Prompting-Review from Last Time**

- Terminology: zero-shot, few-shot, in-context, etc, prompt characteristics: format, examples, orders

- **Improving and Extending Prompting**

- Searching for good prompts, techniques for continuous/soft prompts, ensembling

- **Intro to Chain-of-Thought**

- Basic idea, zero-shot and few-shot, choosing examples for few-shot, tree-of-thoughts

Chain-of-Thought

Performing complex reasoning is hard. Help the model:

Standard Prompting

Model 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

Model 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: Applications

Some things it can be used for:

- **Math problems**
- **Commonsense reasoning**
- **Symbolic reasoning**

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.

Chain-of-Thought: Zero-Shot

No examples shown; encourage model to decompose

- Add to prompt: “Let’s think step by step” before each answer
- For answer extraction, add prompts like “Therefore, the answer (arabic numerals) is” (Kojima et al ‘23)

(a) Few-shot

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: 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:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

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: 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:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(c) Zero-shot

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 ✗

(d) Zero-shot-CoT (Ours)

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?

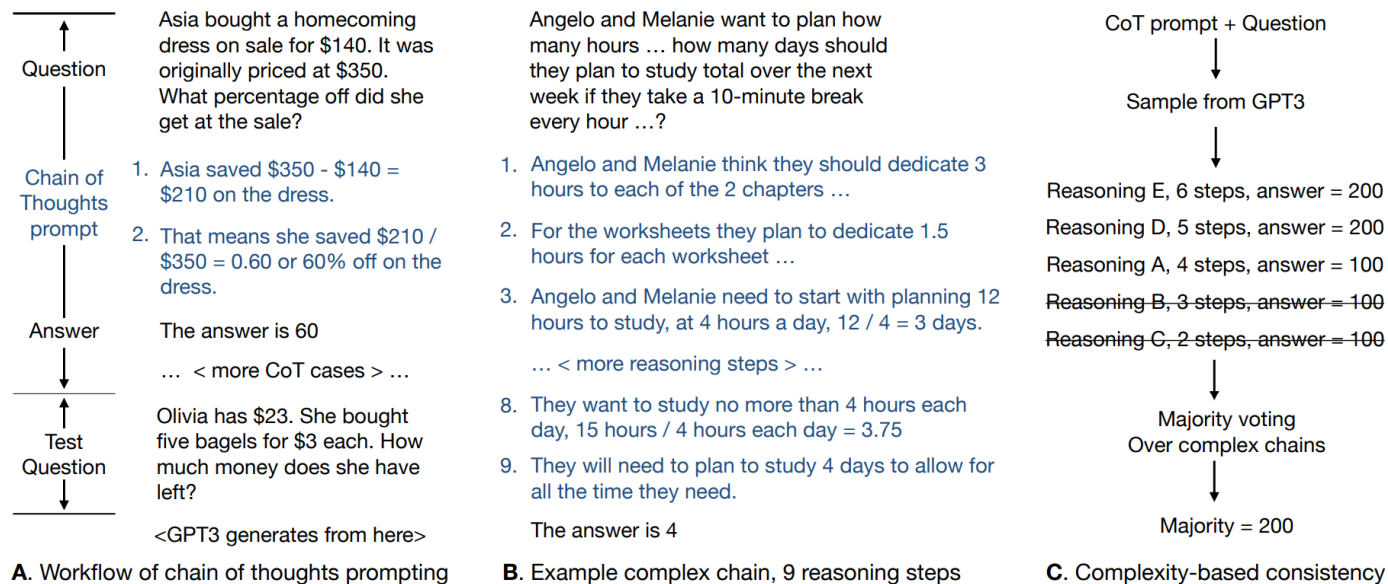
A: **Let's think step by step.**

(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. ✓

Chain-of-Thought: Few-Shot Examples

As before, we must choose few-shot examples.

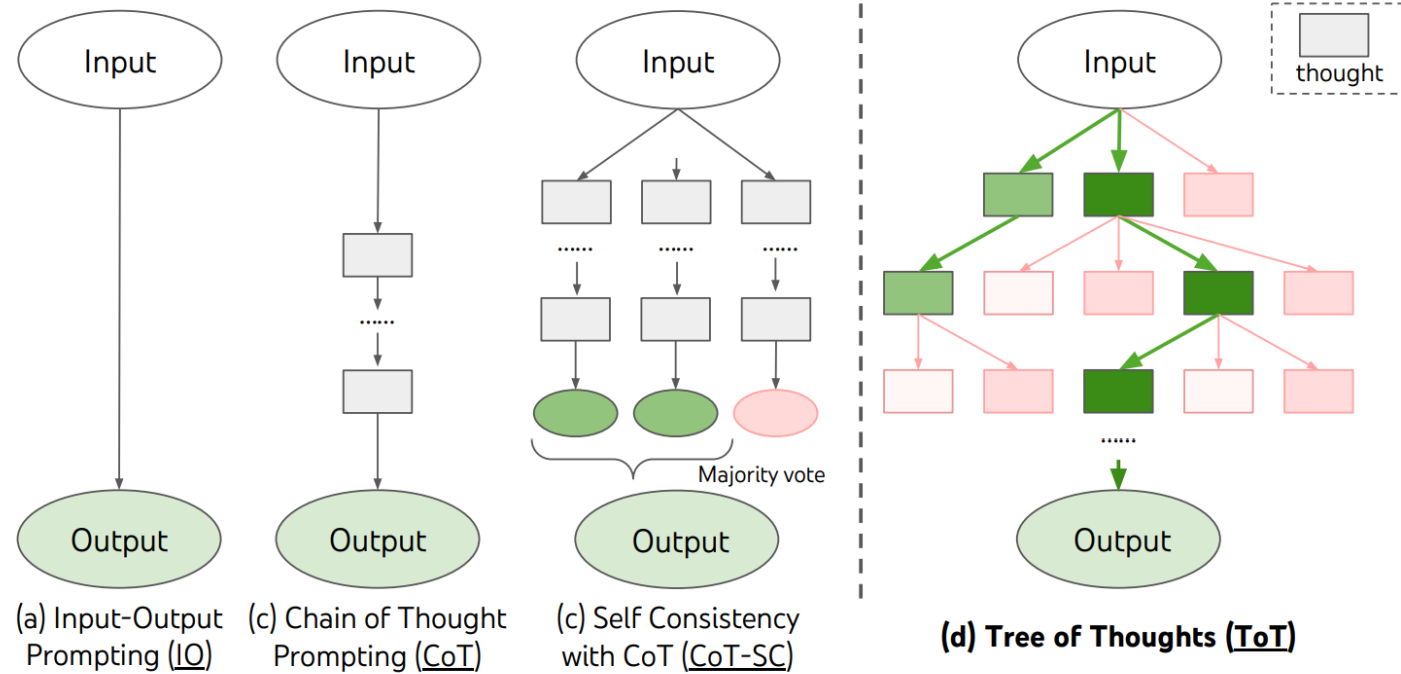
- More structured than simple semantic similarity
- *Complexity-based* prompting.
 - “[S]imply choose complex prompts over simple ones.”
- Prompting: include most steps. Ensembling: MV over set of most complex chains.



Chain-of-Thought: Generalizations

How do we really “reason”?

- Not really by sampling a bunch of chains...



Yao et al '23

Bibliography

- Zhao et al '21: Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, Sameer Singh, “Calibrate Before Use: Improving Few-Shot Performance of Language Models” (<https://arxiv.org/abs/2102.09690>)
- Dong et al '23: Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, Zhifang Sui, “A Survey on In-context Learning” (<https://arxiv.org/abs/2301.00234>)
- Zhou et al '23: Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, Jimmy Ba, “Large Language Models Are Human-Level Prompt Engineers” (<https://arxiv.org/abs/2211.01910>)
- Yang et al '23: Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, Xinyun Chen, “Large Language Models as Optimizers” (<https://arxiv.org/abs/2309.03409>)
- Yuksekgonul et al '24: Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, James Zou, “TextGrad: Automatic "Differentiation" via Text” (<https://arxiv.org/abs/2406.07496>)
- Menon and Vondrick '23, “Visual Classification via Description from Large Language Models” (<https://arxiv.org/abs/2210.07183>)
- Adila '23: Dyah Adila, Changho Shin, Linrong Cai, Frederic Sala, “Zero-Shot Robustification of Zero-Shot Models With Foundation Models” (<https://arxiv.org/pdf/2309.04344.pdf>)
- Gal et al '22: Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, Daniel Cohen-Or, “An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion” (<https://arxiv.org/abs/2208.01618>)
- Zhang et al '23: Yuanhan Zhang, Kaiyang Zhou, Ziwei Liu, “What Makes Good Examples for Visual In-Context Learning?” (<https://arxiv.org/abs/2301.13670>)
- Wei et al '22: Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, Denny Zhou, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models” (<https://arxiv.org/abs/2201.11903>)
- Kojima et al '23: Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, Yusuke Iwasawa, “Large Language Models are Zero-Shot Reasoners” (<https://arxiv.org/abs/2205.11916>)
- Fu et al '23: Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, Tushar Khot, “Complexity-Based Prompting for Multi-Step Reasoning” (<https://arxiv.org/abs/2210.00720>)
- Yao et al '23: Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, Karthik Narasimhan, “Tree of Thoughts: Deliberate Problem Solving with Large Language Models” (<https://arxiv.org/abs/2305.10601>)



Thank You!