



CS 329X: Human Centered LLMs

The Ultimate Crash into LLMs

Diyi Yang, Yijia Shao

Announcements

- Waitlist
- Compute credits
- Homework 1

Available Resources at Stanford

- Farmshare: <https://docs.farmshare.stanford.edu/>
 - 4 rice servers (login nodes) along with 26 compute nodes (iron, rye, wheat, oat nodes)
- Stanford Claude: <https://uit.stanford.edu/service/claude>
- Stanford GPT: <https://uit.stanford.edu/service/openai-chatgpt-edu>

Homework 1

▾ Assignments



HW 1: Model Alignment and Human Feedback

Not available until Sep 28 at 5:45pm | Due Oct 12 at 11:59pm | -/8 pts

- Release: After class on Canvas
- Deadline: Oct 12 11:49 PM PT
- Total: 8 points
 - 3 points for grading the submission + 5 points for the timed quiz
- Use AI & Search to unblock yourself (e.g., Hugging Face tokens, Colab setup, debugging code, etc.), not to replace your understanding
- Timed quiz: One attempt only – take the quiz when you're prepared!

Outline

**YIJIA KNOWS SHE NEED TO
LEACH LLM IN 80 MINUTES**



Outline

- **Transformers and Large Language Models** (30 mins)
- **Prompting** (20 mins)
- **Optimization and Calibration** (20 mins)

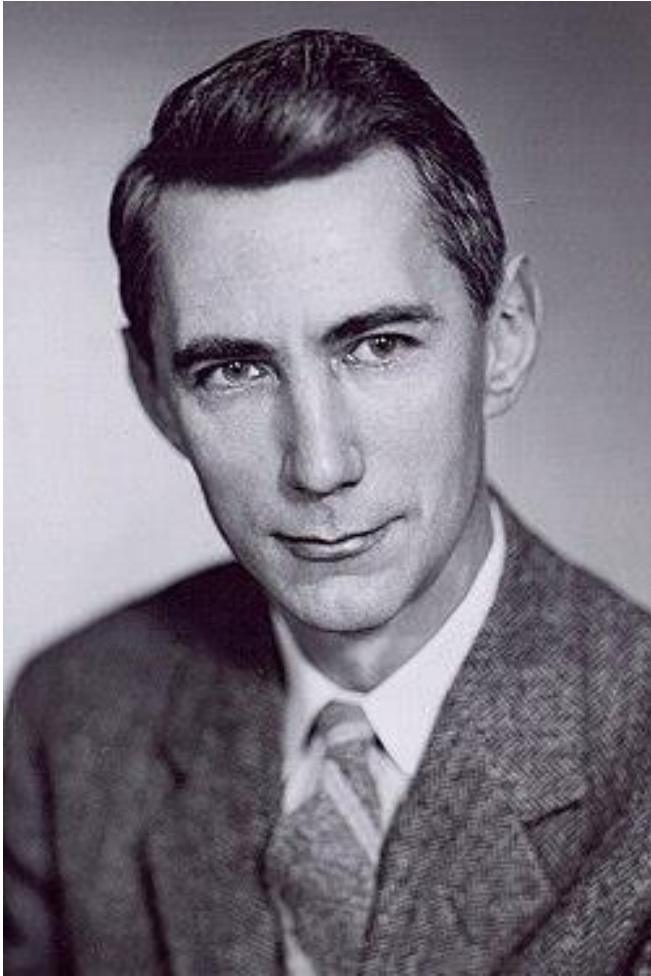
Learning Objective: understand different prompting strategies; learn how to use, optimize, and reflect on their effectiveness and sensitivities

Outline

➤ **Transformers and Large Language Models** (30 mins)

- Language models
- Transformers
- Pretraining and fine-tuning
- ChatGPT & Learning from human preferences
- Emerging topics in LLMs

How To Model Language?



Claude Shannon

A Mathematical Theory of Communication

By C. E. SHANNON

“We may consider a discrete source to be represented by a **stochastic process**.” --1948

$$\{X_t\}_{t \in T}$$

Prediction and Entropy of Printed English

By C. E. SHANNON

(Manuscript Received Sept. 15, 1950)

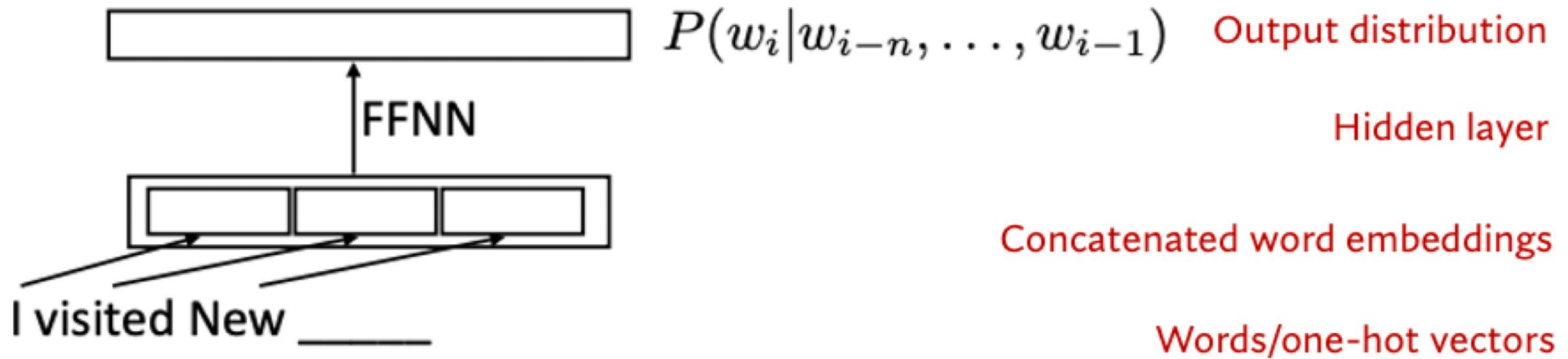
(1) THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG

(2) ———R00———NOT-V———I. SM———OBL———

Language Modeling

- **Input:** sequence of words
- **Output:** probability of the next word

Early work: feedforward neural networks looking at context



Language Modeling Evaluation

- Accuracy doesn't make sense
- Predicting the next word is generally impossible so accuracy would be very low
- A smoother objective (average log-likelihood per token):

$$\frac{1}{n} \sum_{i=1}^n \log P(w_i | w_1, \dots, w_{i-1})$$

Training Objective:

$$\max_{\theta} \frac{1}{n} \sum_{i=1}^n \log P(w_i | w_1, \dots, w_{i-1})$$

Evaluation Objective (Perplexity):

$$PPL(w_{1:n}) = \exp\left(-\frac{1}{n} \sum_{i=1}^n \log P(w_i | w_1, \dots, w_{i-1})\right)$$

Statistical Language Modeling

- n-gram Language Models: Predict from counts of short contexts

$$P(w_i | w_{<i}) \approx P(w_i | w_{i-N:i-1})$$

$$P(\text{blue} | \text{The water of Walden Pond is so beautifully}) = \frac{C(\text{The water of Walden Pond is so beautifully blue})}{C(\text{The water of Walden Pond is so beautifully})}$$

Approximate with bigram (N=2)

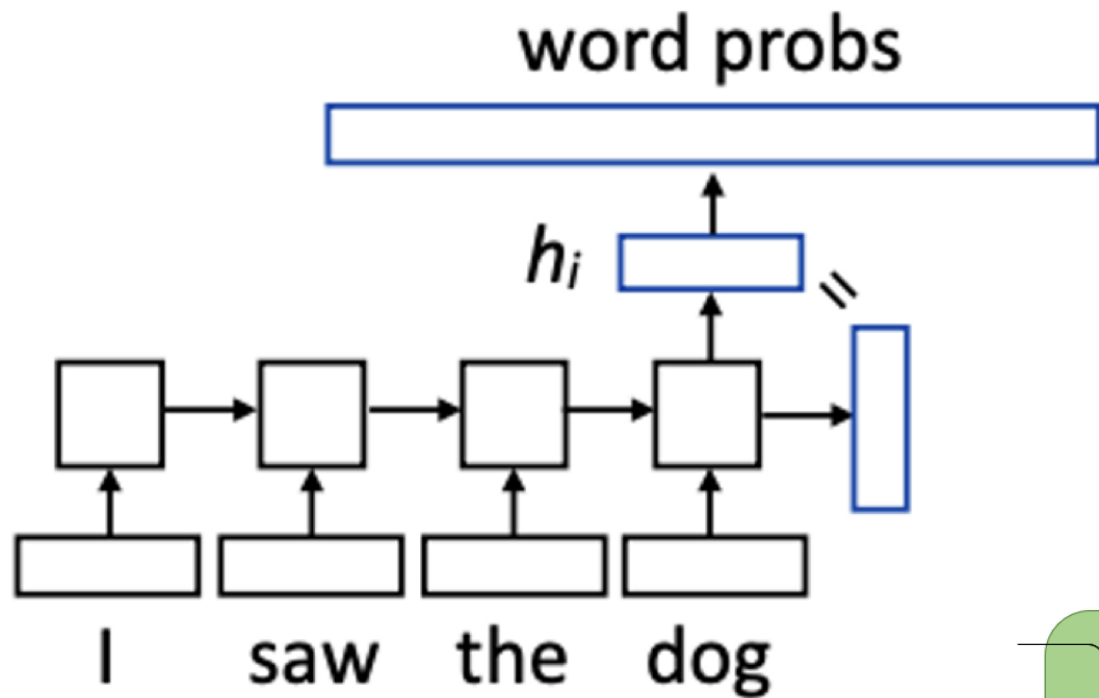
$$P(\text{blue} | \text{beautifully}) = \frac{C(\text{beautifully blue})}{C(\text{beautifully})}$$

Approximate with trigram (N=3)

$$P(\text{blue} | \text{so beautifully}) = \frac{C(\text{so beautifully blue})}{C(\text{so beautifully})}$$

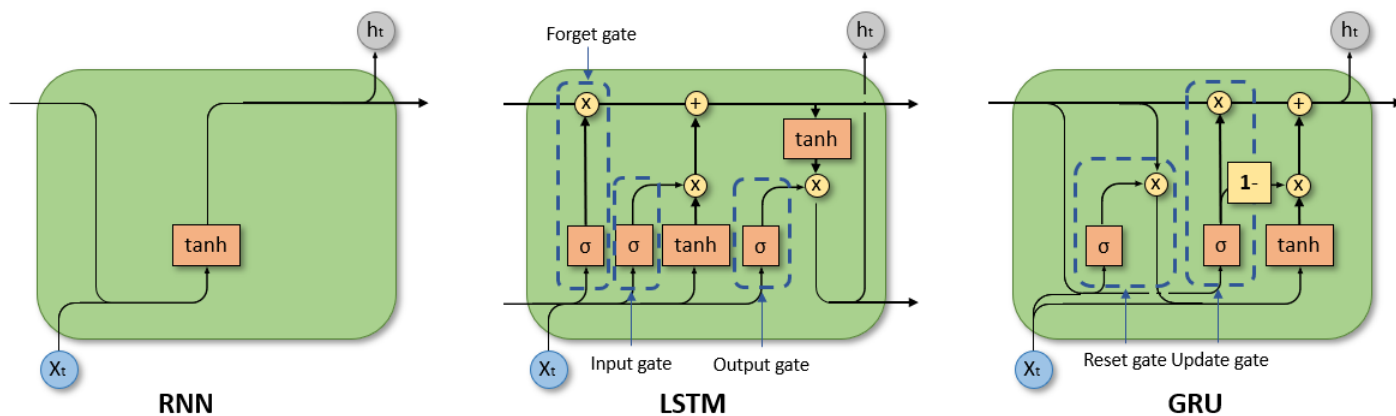
- Neural Language Models: Can we **learn a representation of context** instead of counting every context?

Language Modeling via Recurrent Neural Network



$$P(w|\text{context}) = \text{softmax}(W\mathbf{h}_i)$$

W is a (vocab size) x (hidden size) matrix



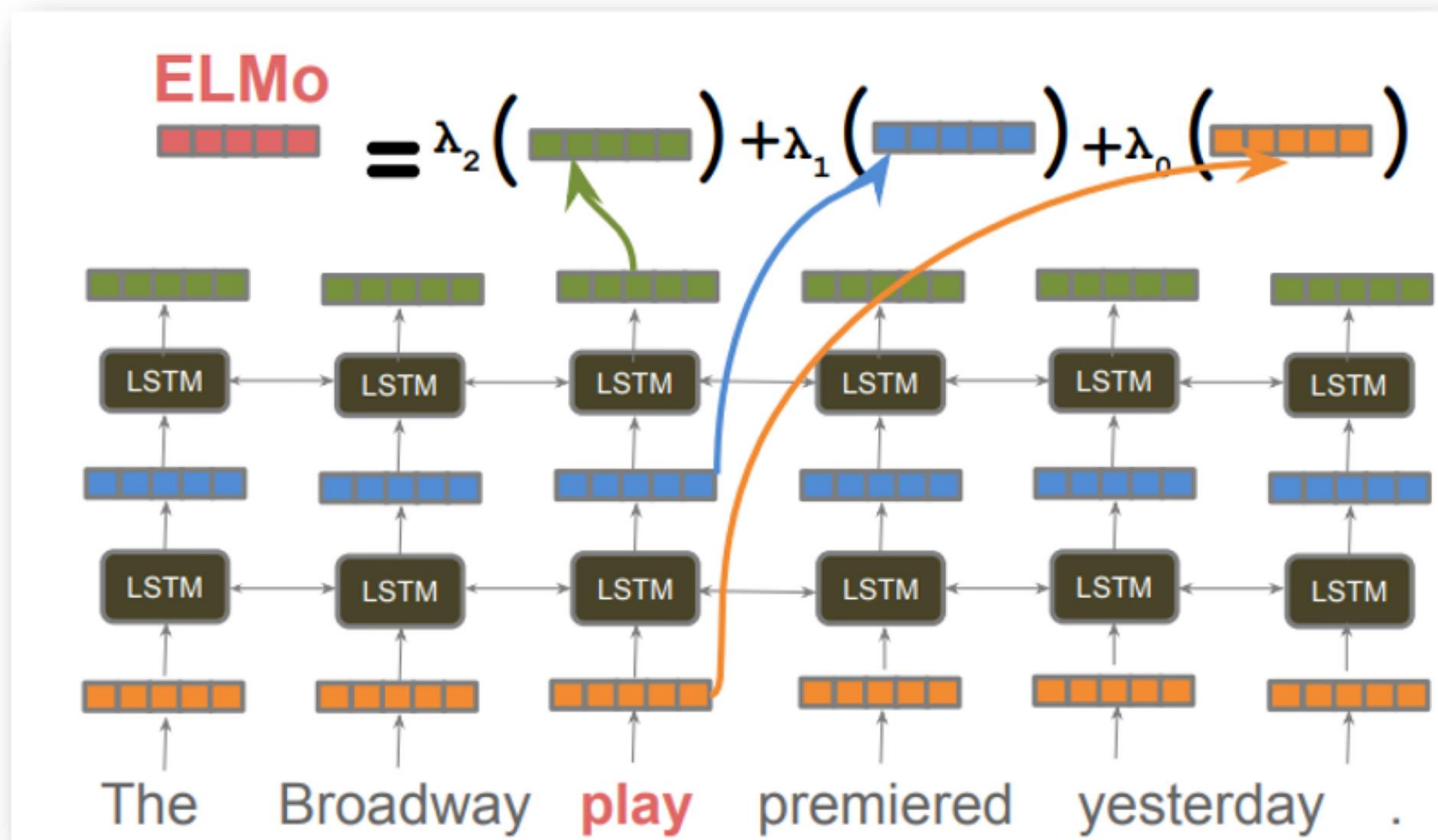
ELMO

Deep contextualized word representations

Matthew E. Peters[†], Mark Neumann[†], Mohit Iyyer[†], Matt Gardner[†],
{matthewp, markn, mohiti, mattg}@allenai.org

Christopher Clark^{*}, Kenton Lee^{*}, Luke Zettlemoyer^{†*},
{csquared, kentonl, lsz}@cs.washington.edu

Published in NAACL 2018



Limitations of RNN LMs

- They can't remember earlier words and can't go back and forth
- Need pointing mechanisms to repeat recent words
- Transformers can help!

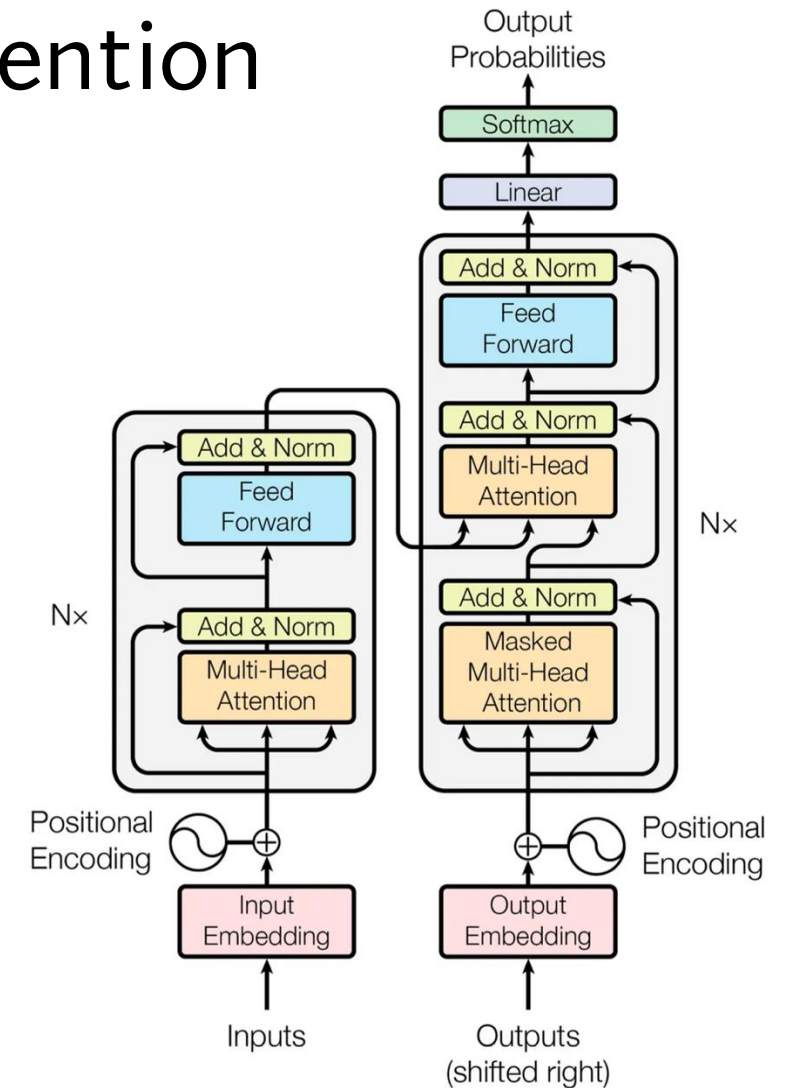
Recurrent models and attention

- Use attention to allow flexible access to memory
- Attention treats each word's representation as a query to access and incorporate information from a set of values.
- Instead of attention from the decoder to the encoder, Transformer operationalizes attention **within a single sentence**.

Transformer with Multi-headed Attention

Benefits of Transformers:

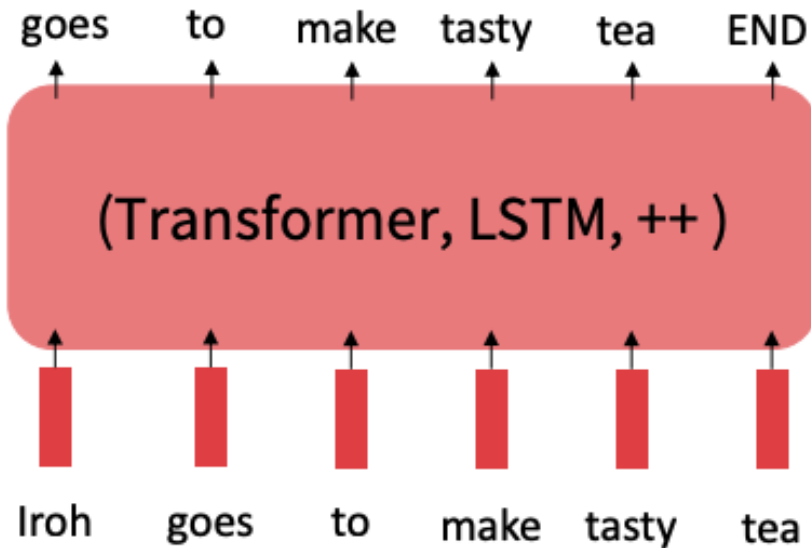
- Capture long- and short-term dependencies
- Efficient backpropagation
- Parallelizable
- Allow deeper architectures
- Allow multimodality (image, speech, text ...)



Pretraining and Finetuning Paradigm

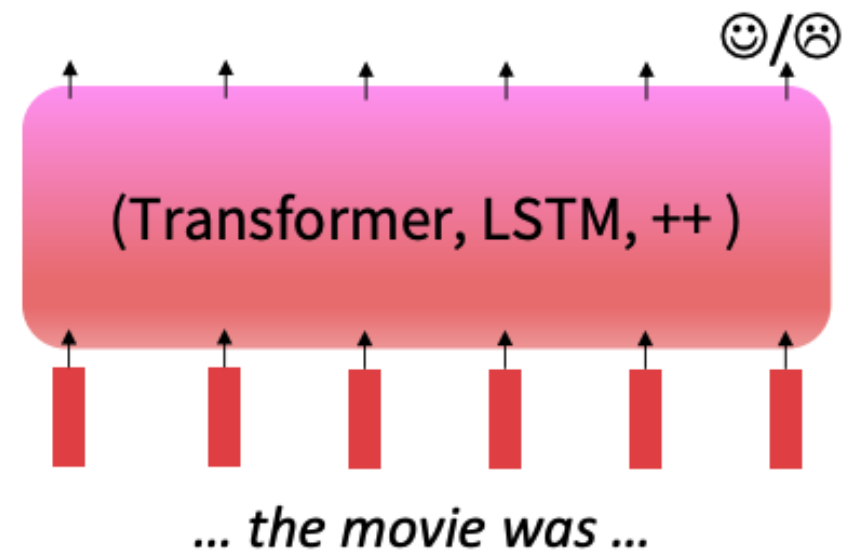
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on your task)

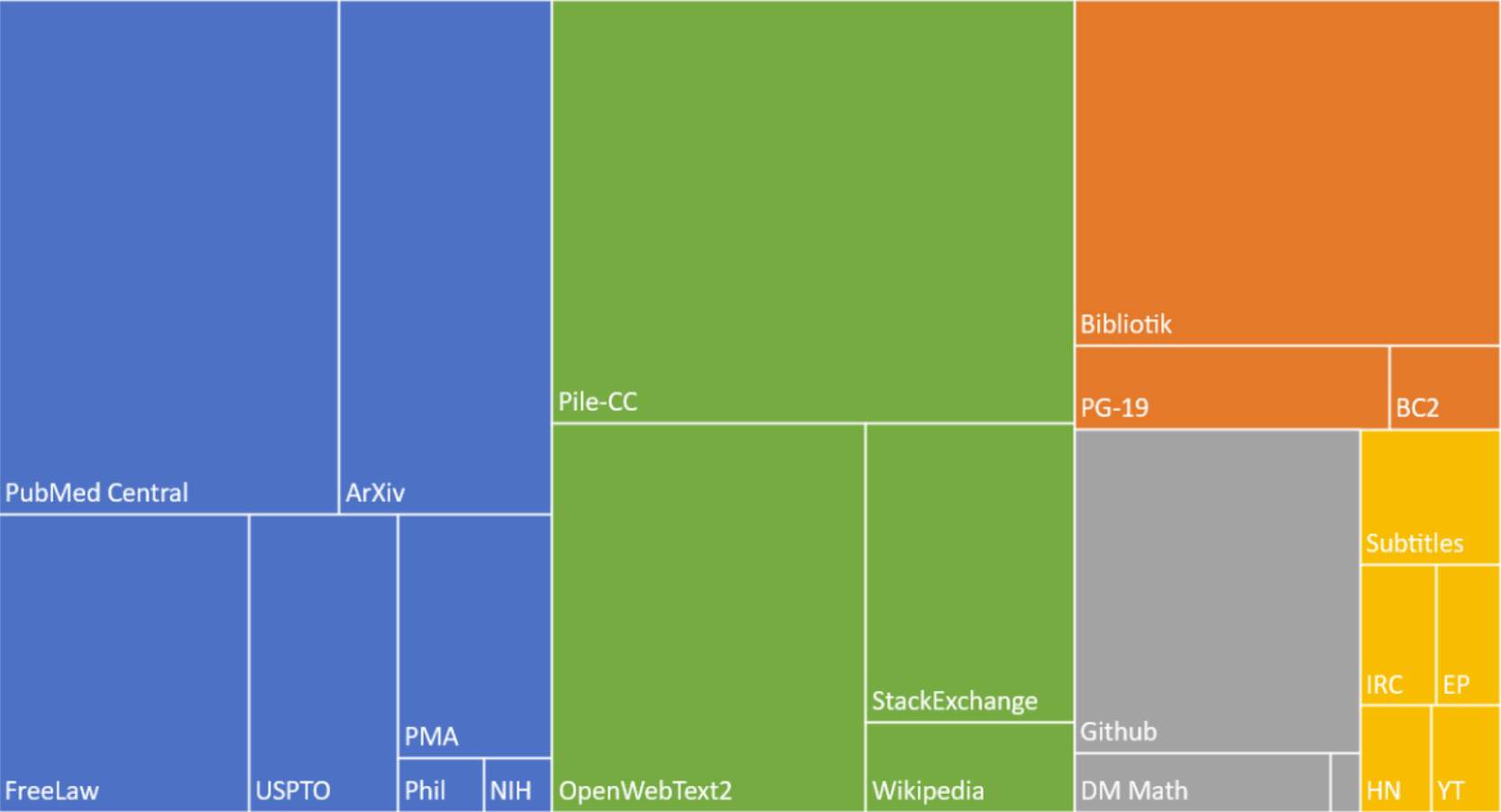
Not many labels; adapt to the task!



Where does this data come from?

Composition of the Pile by Category

■ Academic ■ Internet ■ Prose ■ Dialogue ■ Misc

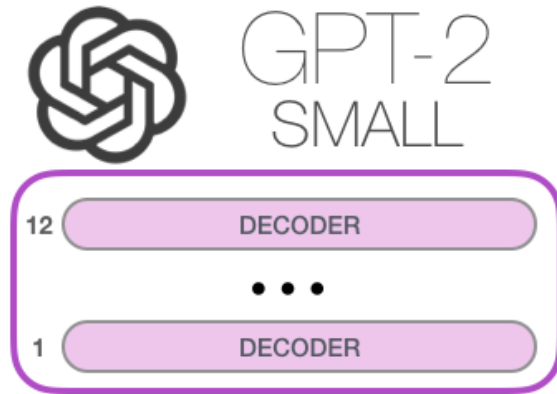


Model	Training Data
BERT	BookCorpus, English Wikipedia
GPT-1	BookCorpus
GPT-3	CommonCrawl, WebText, English Wikipedia, and 2 book databases (“Books 1” and “Books 2”)
GPT-3.5+	Undisclosed

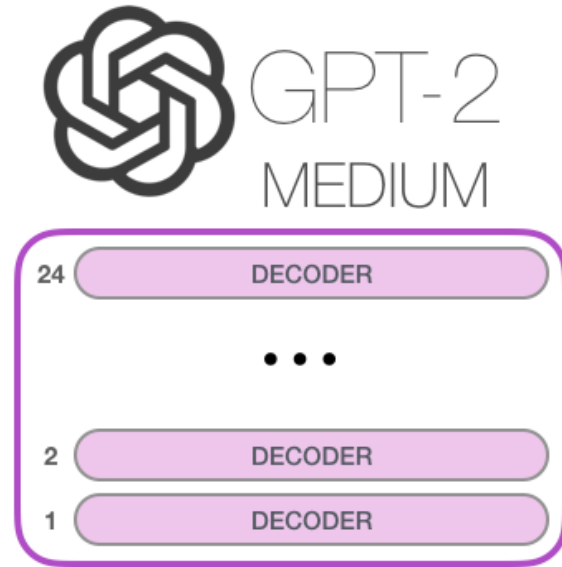
Generative Pretrained Transformer (GPT) ([Radford et al., 2018](#))

- 2018's GPT was a big success in pretraining a decoder!
- Transformer decoder with 12 layers, 117M parameters
 - 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers.
 - Byte-pair encoding
- Trained on BooksCorpus: over 7000 unique books
 - Contains long spans of contiguous text, for learning long-distance dependencies

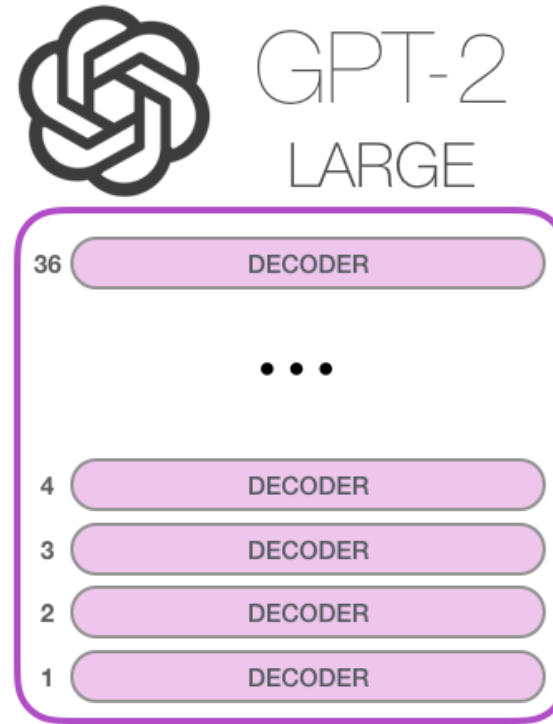
GPT-2 (Larger version of GPT)



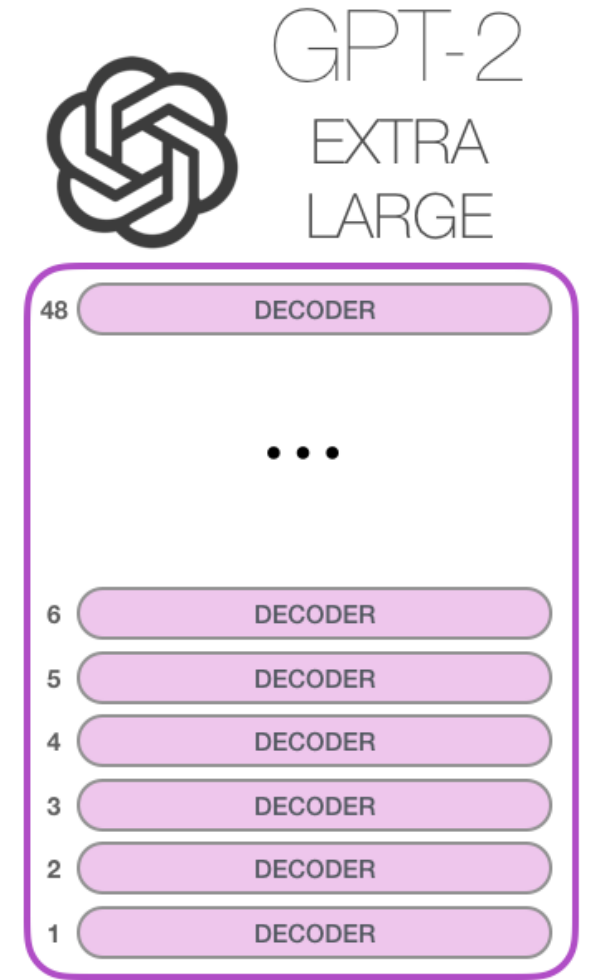
Model Dimensionality: 768



Model Dimensionality: 1024



Model Dimensionality: 1280



Model Dimensionality: 1600

<https://jalammar.github.io/illustrated-gpt2/>

More is different: **large** language models

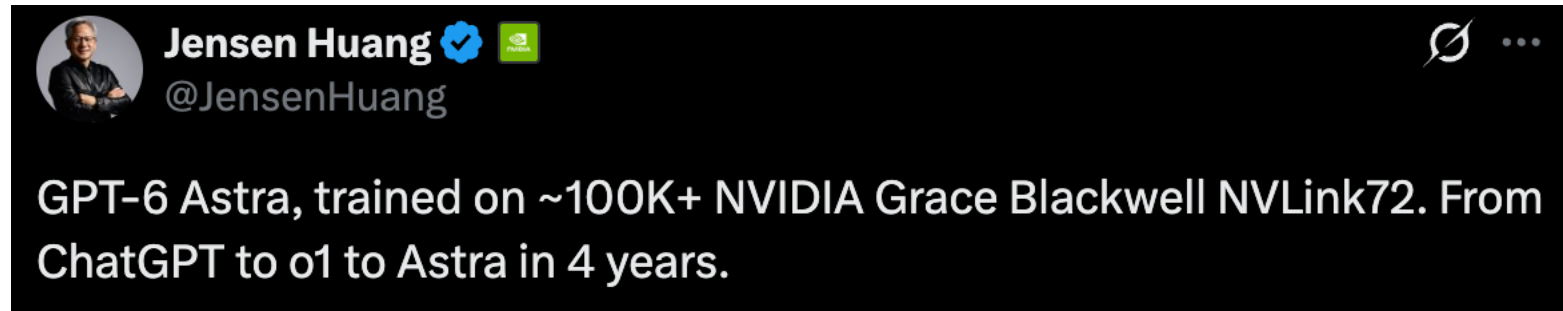
GPT-1: 12 layers, 12 heads, 120M parameters, 4.5GB training data.

GPT-2: 48 layers, 25 heads, 1.5B parameters, 40GB training data.

GPT-3: 96 layers, 128 heads, 175B parameters, 570GB training data.

- Trained by a supercomputer developed by Microsoft Azure.
- 285,000 CPU cores and 10,000 GPUs.
- 400 Gbps of network connectivity for each GPU server.
- The project's estimated cost: 4.6 million.
- Received criticism for the environmental impact for the first time.

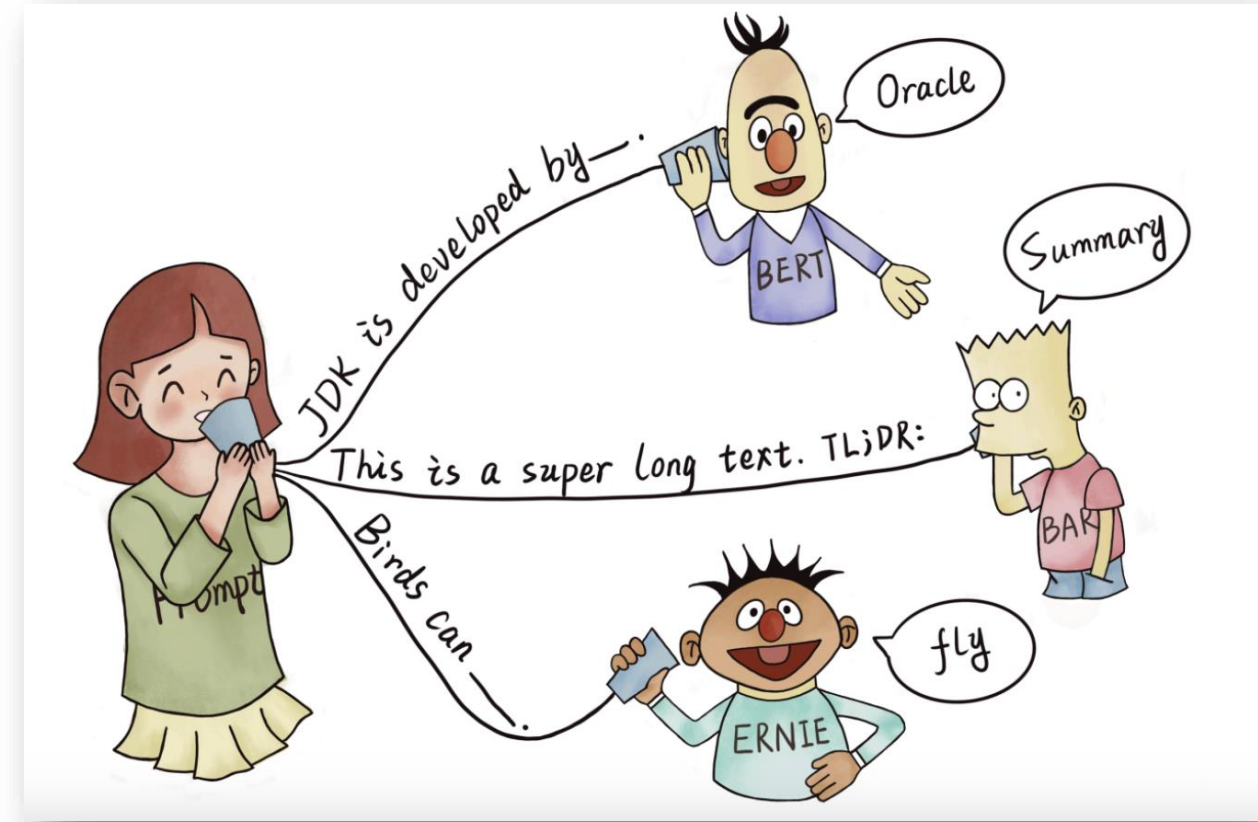
More is different: **large** language models



Prompt for LLMs

Fine-tuning GPT-3 175B in 2020 was not feasible due to its large size

Prompts (or **in-context learning**) were then introduced and used

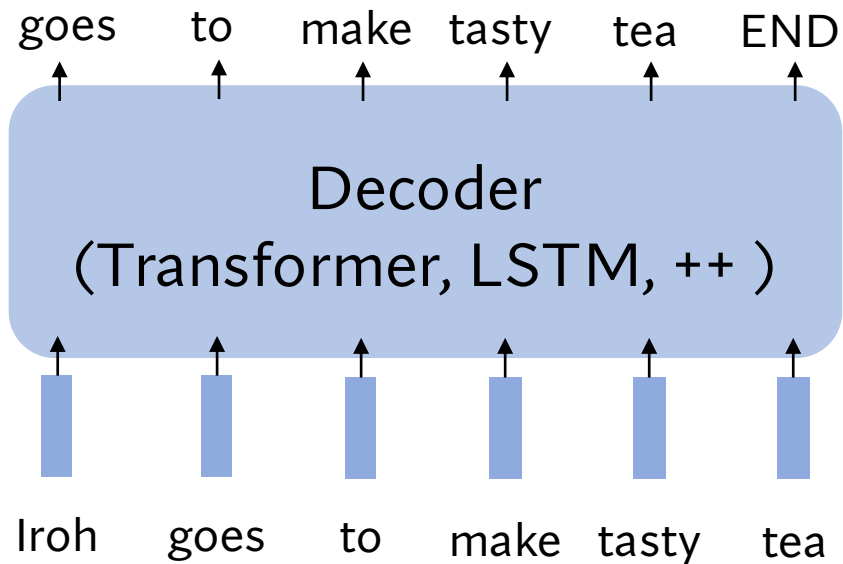


Scaling up finetuning

Pretraining can improve NLP applications by serving as parameter initialization.

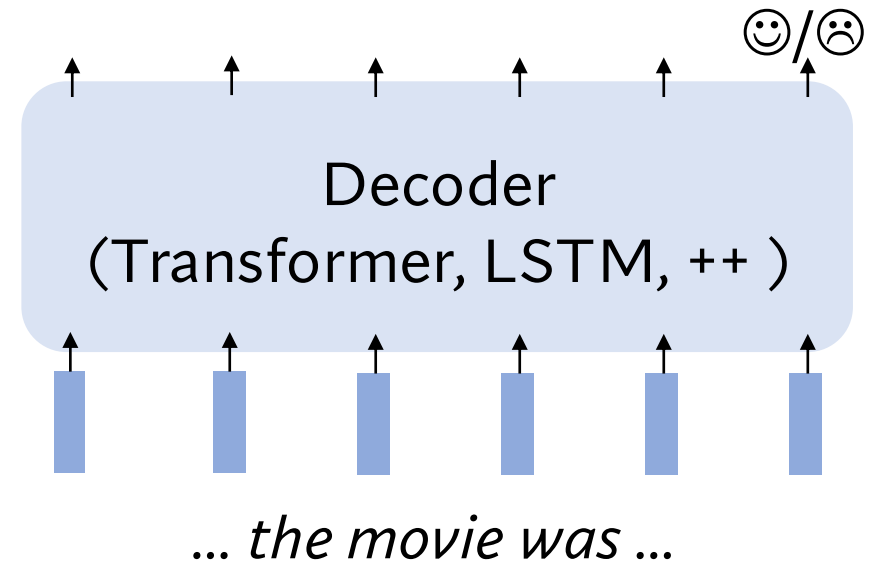
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on **many tasks**)

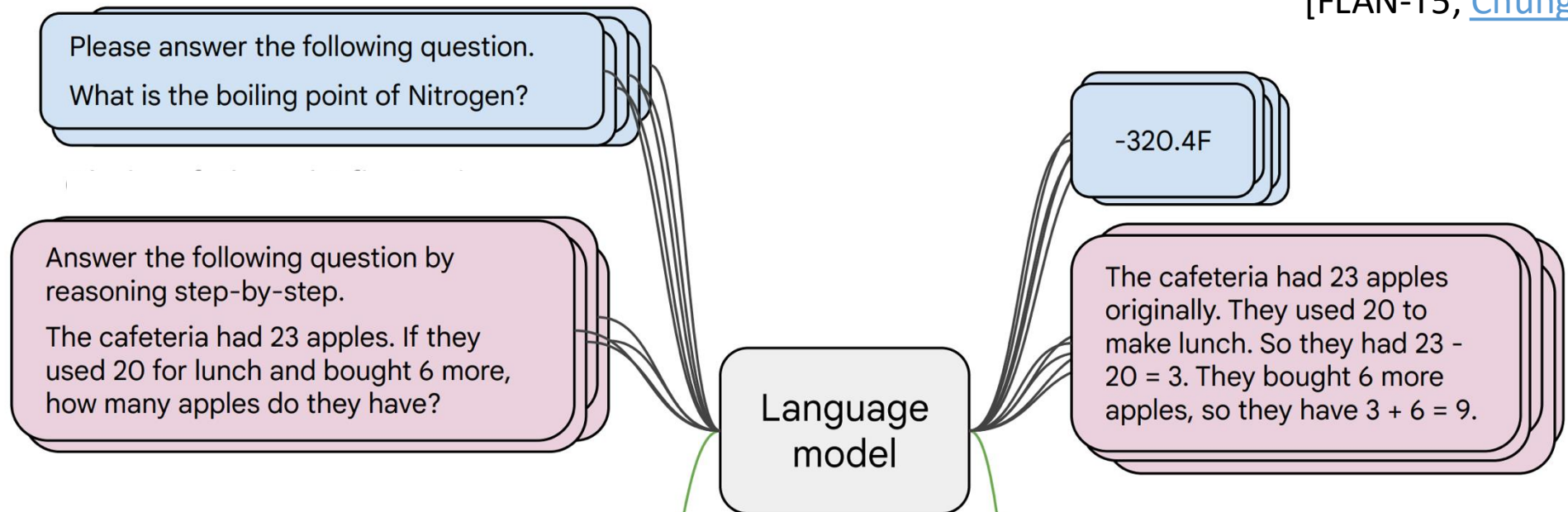
Not many labels; adapt to the tasks!



Instruction finetuning

Collect examples of (instruction, output) pairs across many tasks and finetune an LM

[FLAN-T5; [Chung et al., 2022](#)]



Evaluate on **unseen tasks**

Q: Can Geoffrey Hinton have a conversation with George Washington?
Give the rationale before answering.

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is "no".

Limitations of Instruction Finetuning

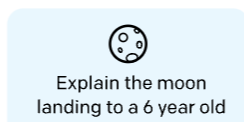
- One limitation of instruction finetuning is obvious: it's **expensive** to collect groundtruth data for tasks.
- **Problem 1:** tasks like open-ended creative generation have no right answer.
- **Problem 2:** language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
- Even with instruction finetuning, there is a **mismatch** between the LM objective and the objective of “satisfy human preferences”!
- Can we explicitly attempt to satisfy human preferences?

Optimizing for Human Preferences: ChatGPT and RLHF

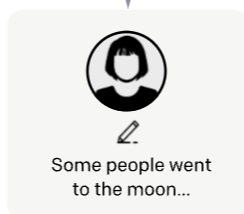
Step 1

Collect demonstration data, and train a supervised policy.

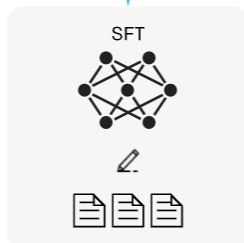
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



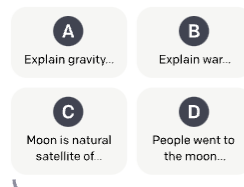
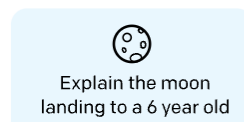
This data is used to fine-tune GPT-3 with supervised learning.



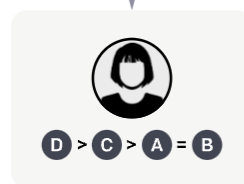
Step 2

Collect comparison data, and train a reward model.

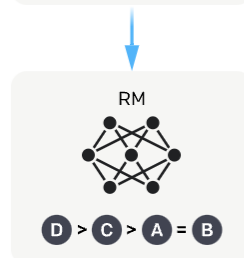
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



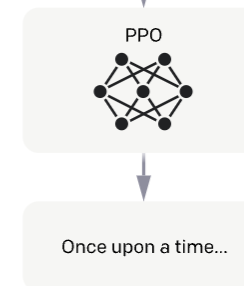
Step 3

Optimize a policy against the reward model using reinforcement learning.

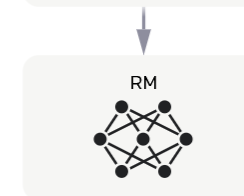
A new prompt is sampled from the dataset.



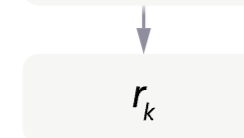
The policy generates an output.



The reward model calculates a reward for the output.



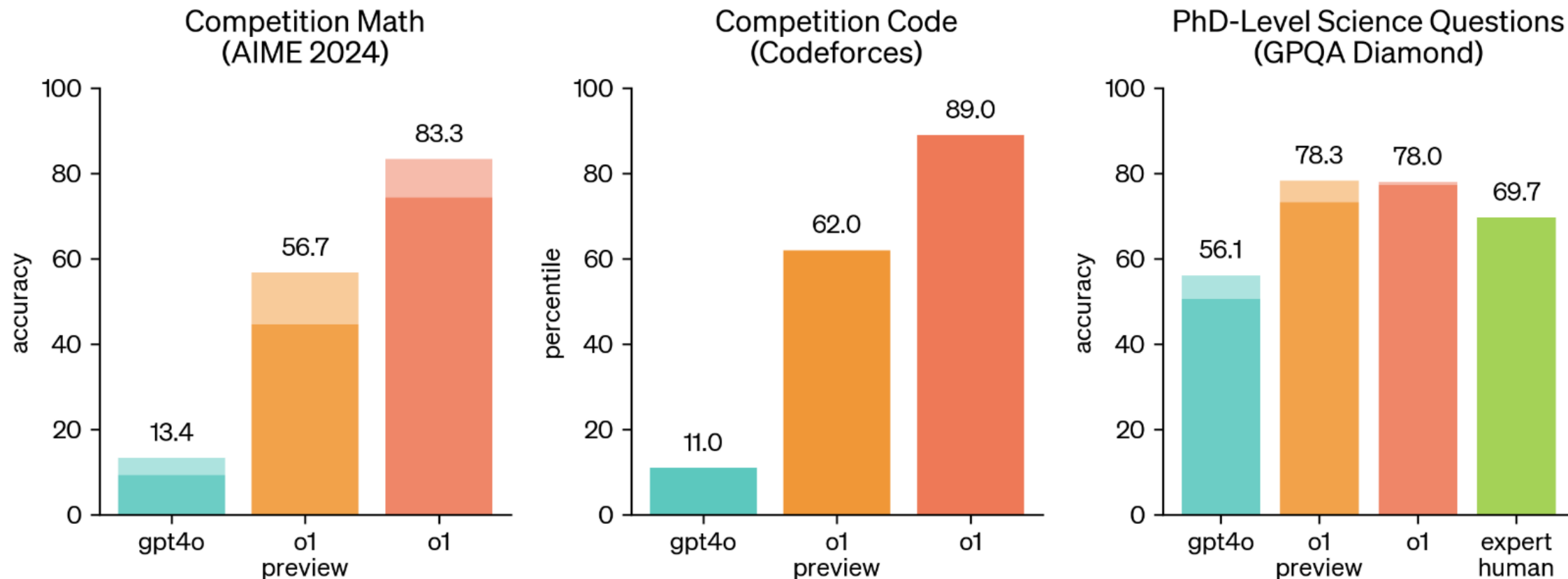
The reward is used to update the policy using PPO.



Limitation of RLHF

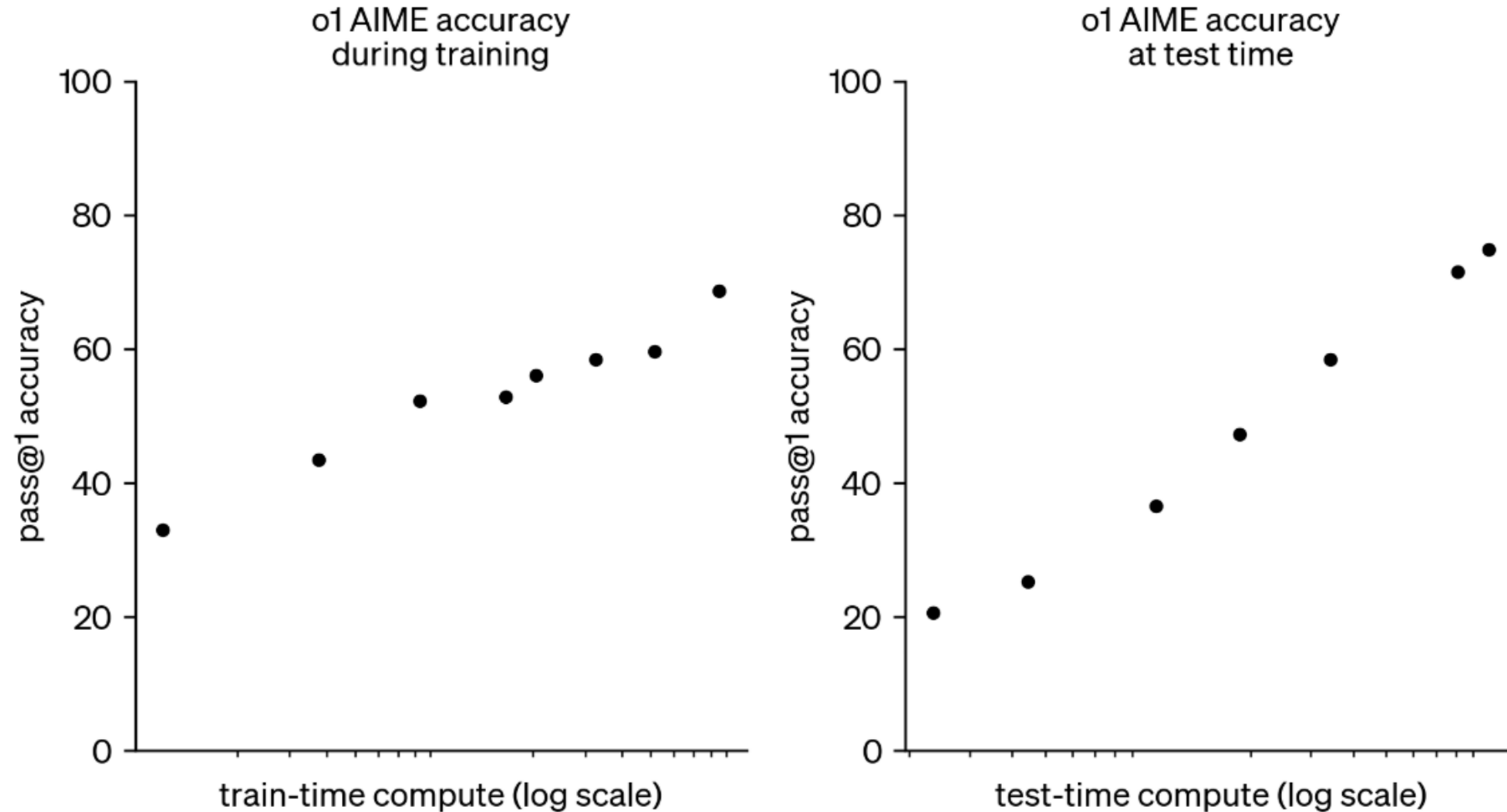
- Human preferences are unreliable
 - Reward hacking is a common problem in RL
 - Chatbots are rewarded to produce responses that seem authoritative and helpful, regardless of truth
 - This can result in making up facts + hallucinations
- RLHF labels are often obtained from overseas, low-wage workers

Reasoning Model O1 with Large-Scale RL

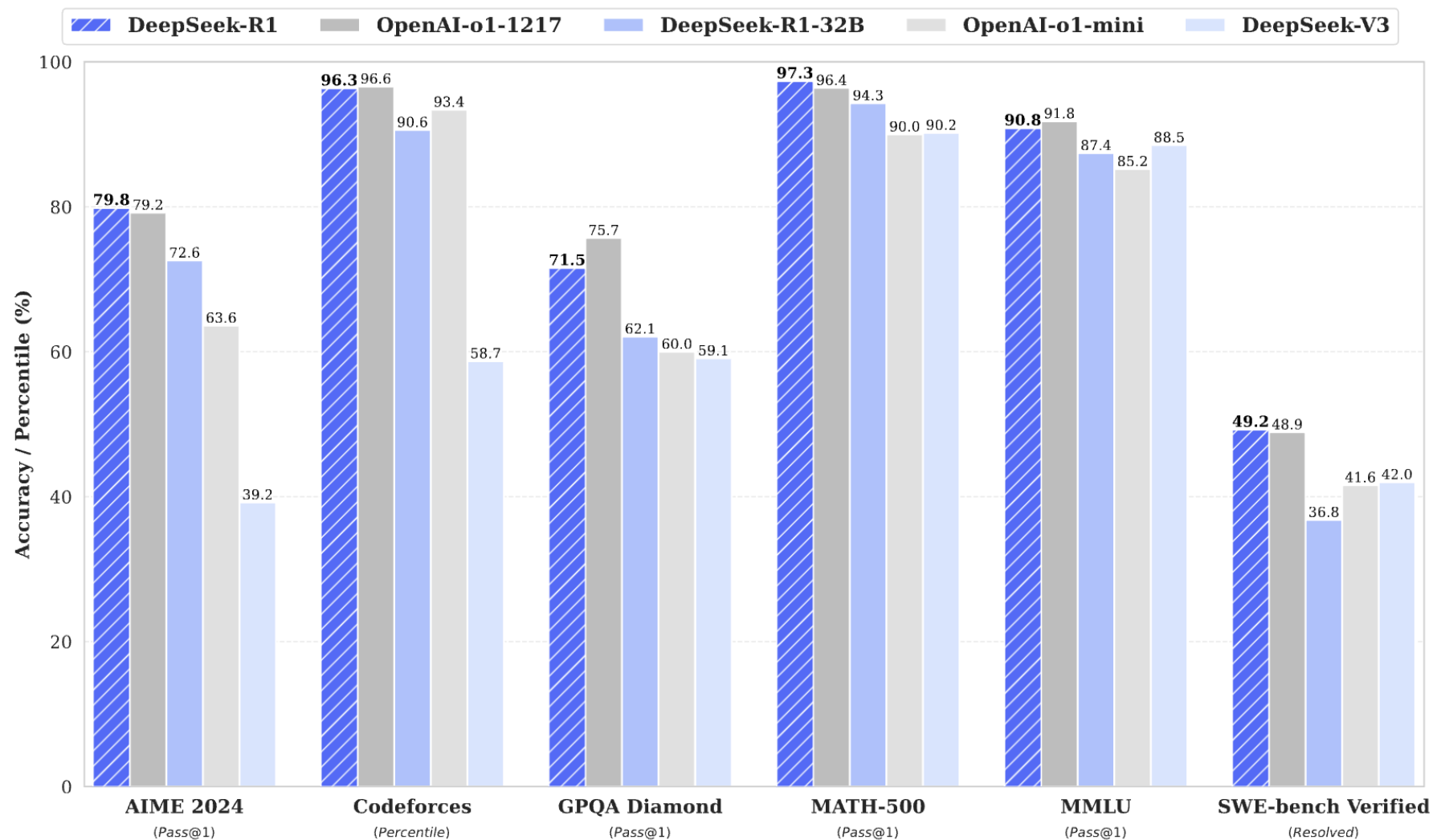


<https://openai.com/index/learning-to-reason-with-llms/>

More thinking (test-time compute) helps a lot



DeepSeek-R1: Smaller & Cheaper but Similar Performance to O1



More Thinking and The “Aha” Moment

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\dots$

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

$\dots$

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be $\dots$

We started with the equation:

$$\sqrt{a - \sqrt{a + x}} = x$$

First, let's square both sides:

$$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$$

Next, I could square both sides again, treating the equation: $\dots$

$\dots$

DeepSeek R1 when it is prompted to select a random number:

Okay, the user wants me to think of a number between 1 and 10,000 but not tell them. Hmm, they mentioned not to overthink it, so maybe just pick a random number quickly. Let me see. Well, people often have favorite numbers or patterns. Maybe go with something like 7777? But that's four 7s, which might be too obvious. Alternatively, 1234? Too sequential. Maybe a number that's a common choice but not too predictable. Wait, I should just pick the first number that comes to mind. Let me try that. Closing my eyes and... 42? Wait, that's from Hitchhiker's Guide, but maybe they don't want a reference. Alternatively, 1004? That's within the range. Or 8675? No, that's from a song. Hmm, maybe 3141? Like the first four digits of pi. But 3141 is 3.141... multiplied by 1000. That's clever, but maybe too nerdy. Alternatively, 1984? But that's a book title. Maybe 2023, the current year? But they might not want a recent year. Or 1337, which is leet speak. But that's only 4 digits. Wait, ...

Credit to <https://x.com/flowersslop/status/1882241958397067677>

Summary

- Language Modeling objective
- Using neural networks for language modeling
 - From RNN to Transformer
- Pretraining and finetuning paradigm
- Large Language Model (LLM): Scaling up model parameters and data
- From predicting next tokens to completing tasks: Instruction Finetuning
- Optimizing for human preferences: RLHF
- Teach model to think before talk: Training reasoning models with RL

Outline

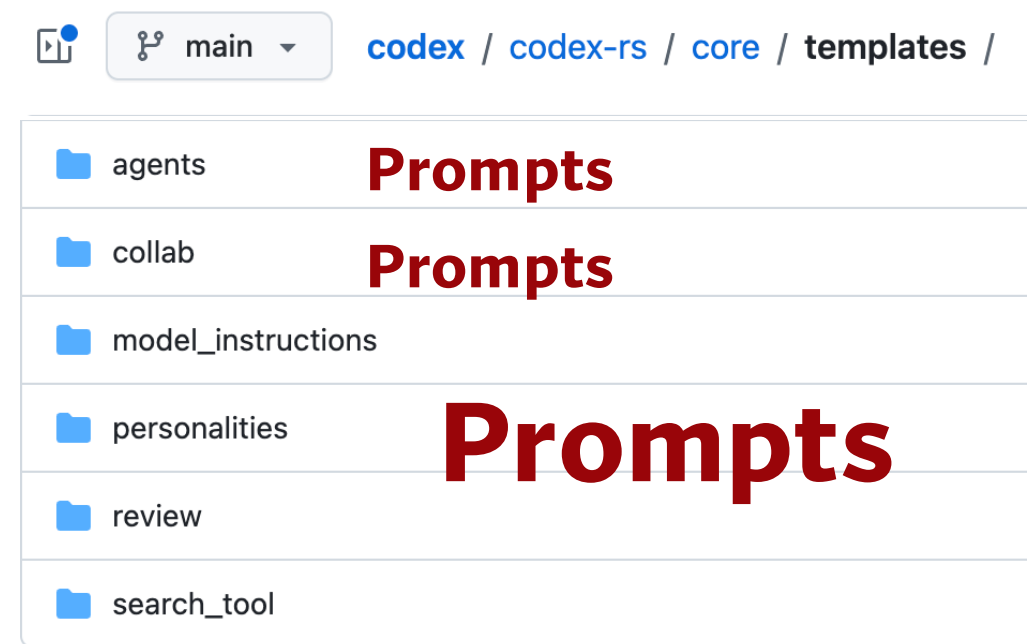
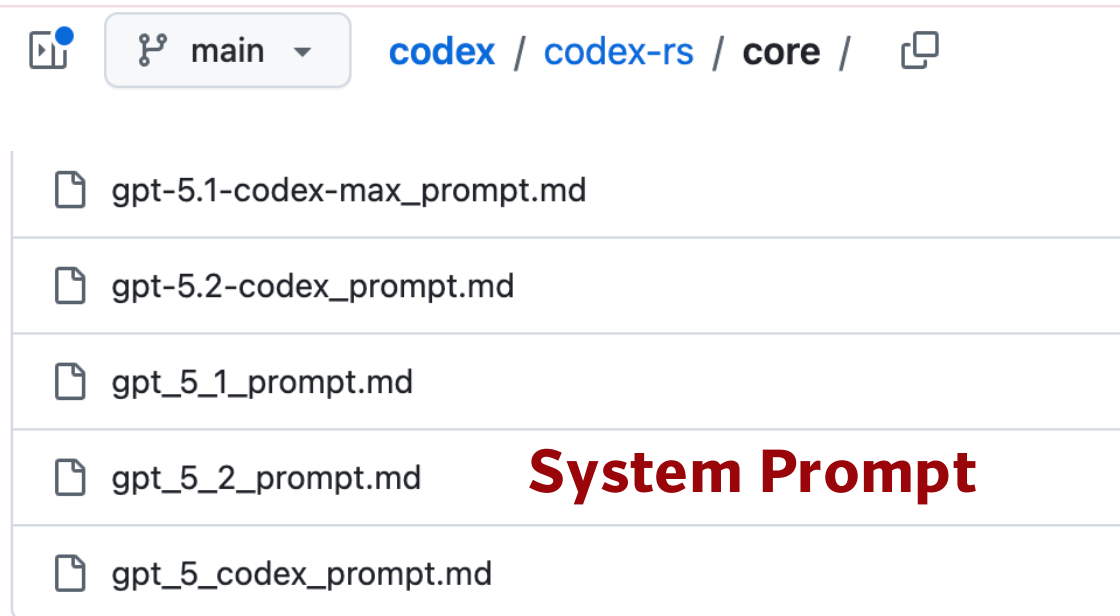
✓ **Transformers and Large Language Models** (30 mins)

➤ **Prompting** (20 mins)



Do We Need To Learn Prompting?

- Everyone can prompt, but good prompting takes skills
- Example: Prompts in Codex (<https://github.com/openai/codex>)



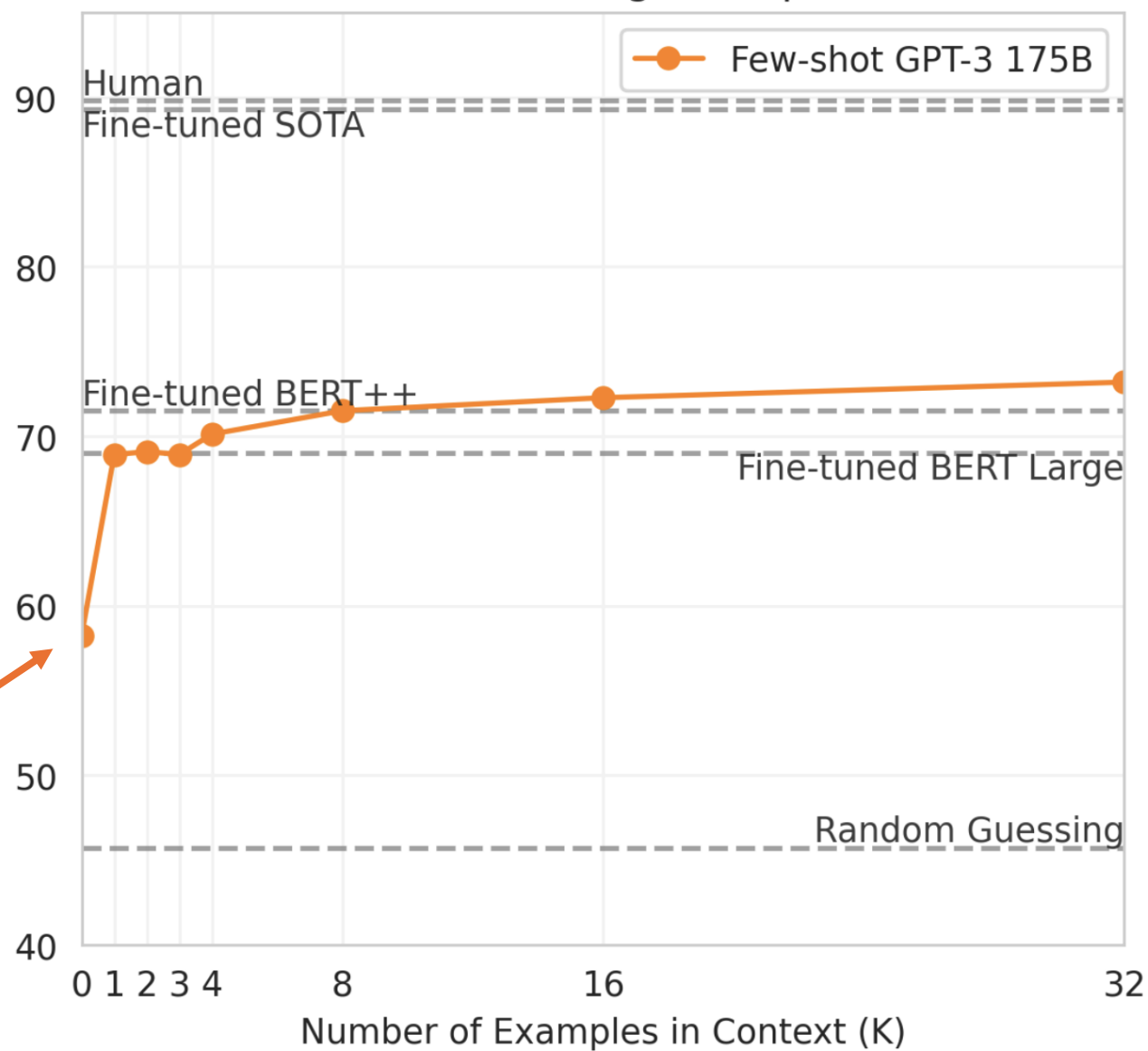
Zero-shot

1 Translate English to French:

2 cheese =>



In-Context Learning on SuperGLUE



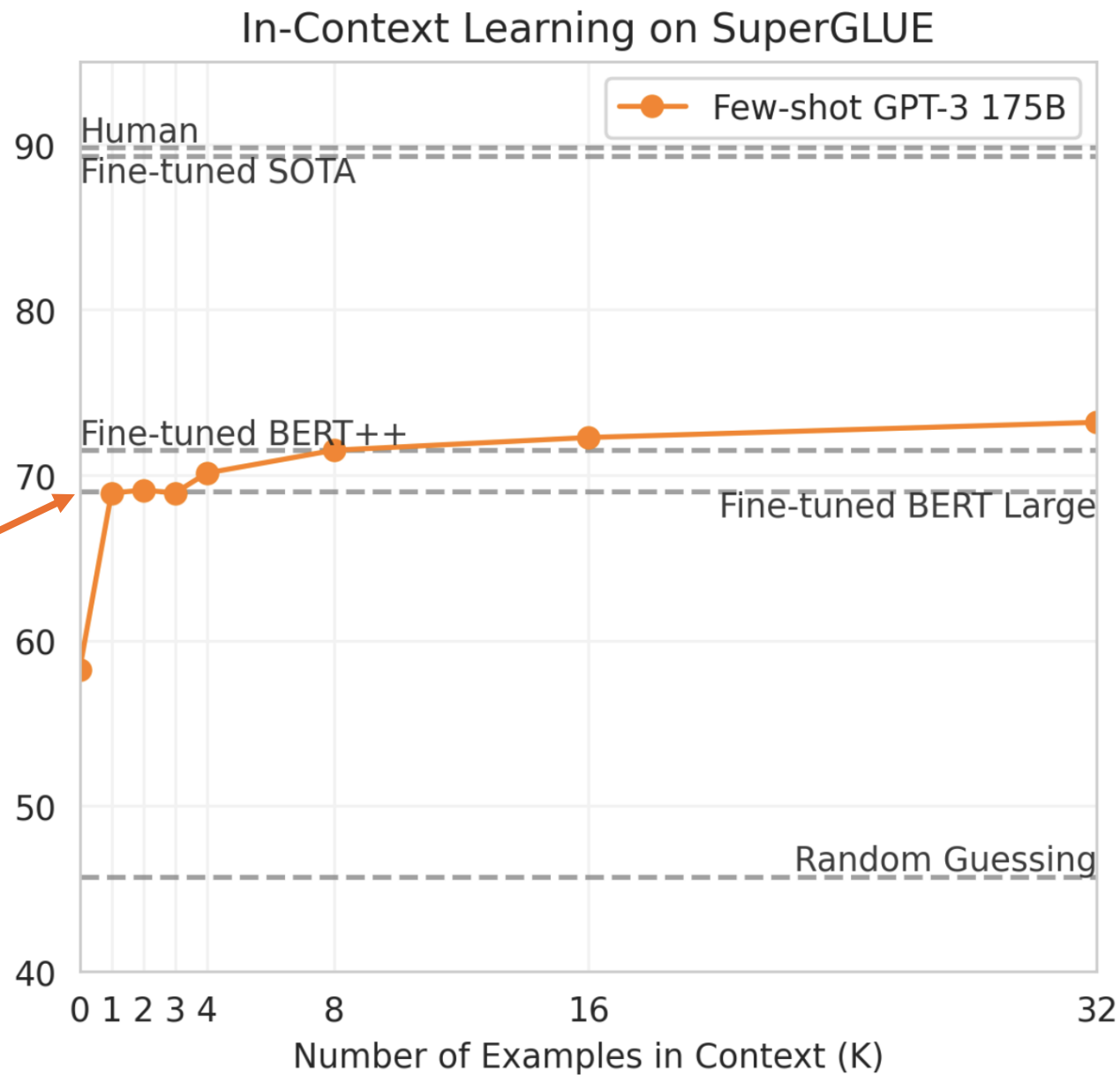
One-shot

1 Translate English to French:

2 sea otter => loutre de mer

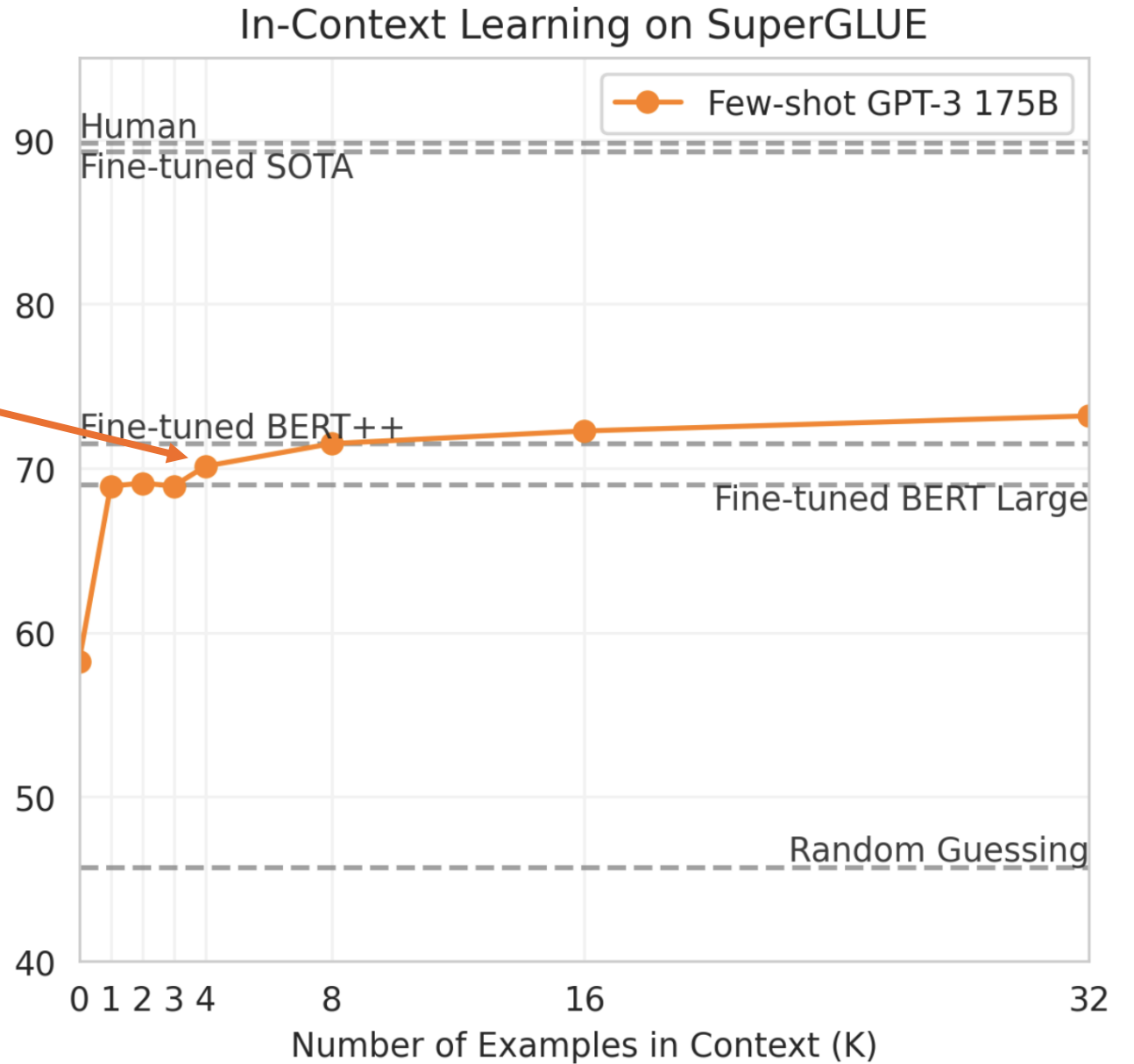
3 cheese =>

.....



Few-shot

1 Translate English to French:
2 sea otter => loutre de mer
3 peppermint => menthe poivrée
4 plush girafe => girafe peluche
5 cheese =>



[[Brown et al., 2020](#)]

Zero/few-shot prompting

```
1 Translate English to French: ←  
2 sea otter => loutre de mer ←  
3 peppermint => menthe poivrée ←  
4 plush girafe => girafe peluche ←  
5 cheese => ..... ←
```

Traditional fine-tuning



[[Brown et al., 2020](#)]

Limits of prompting for harder tasks?

Some tasks seem too hard for even large LMs to learn through prompting alone.
Especially tasks involving **richer, multi-step reasoning**.
(Humans struggle at these tasks too!)

$$19583 + 29534 = 49117$$

$$98394 + 49384 = 147778$$

$$29382 + 12347 = 41729$$

$$93847 + 39299 = ?$$

Solution: change the prompt!

Chain-of-thought prompting

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.

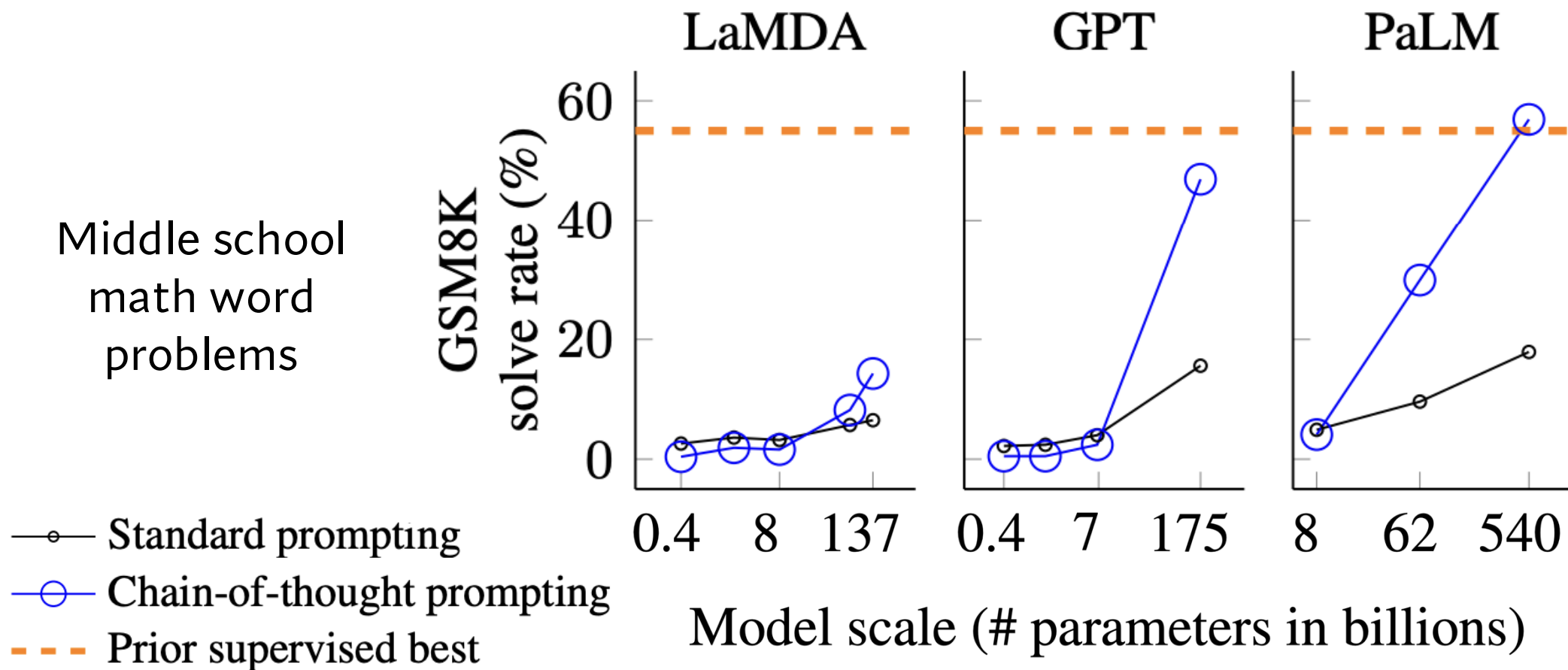
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

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. ✅

[[Wei et al., 2022](#); also see [Nye et al., 2021](#)]

Chain-of-thought prompting is an **emergent property** of model scale



[[Wei et al., 2022](#); also see [Nye et al., 2021](#)]

Zero-shot chain-of-thought prompting

No.	Category	Zero-shot CoT Trigger Prompt	Accuracy
1	LM-Designed	Let's work this out in a step by step way to be sure we have the right answer.	82.0
2	 Human-Designed	Let's think step by step. (*1)	78.7
3		First, (*2)	77.3
4		Let's think about this logically.	74.5
5		Let's solve this problem by splitting it into steps. (*3)	72.2
6		Let's be realistic and think step by step.	70.8
7		Let's think like a detective step by step.	70.3
8		Let's think	57.5
9		Before we dive into the answer,	55.7
10		The answer is after the proof.	45.7
-		(Zero-shot)	17.7

Outline

✓ **Transformers and Large Language Models** (30 mins)

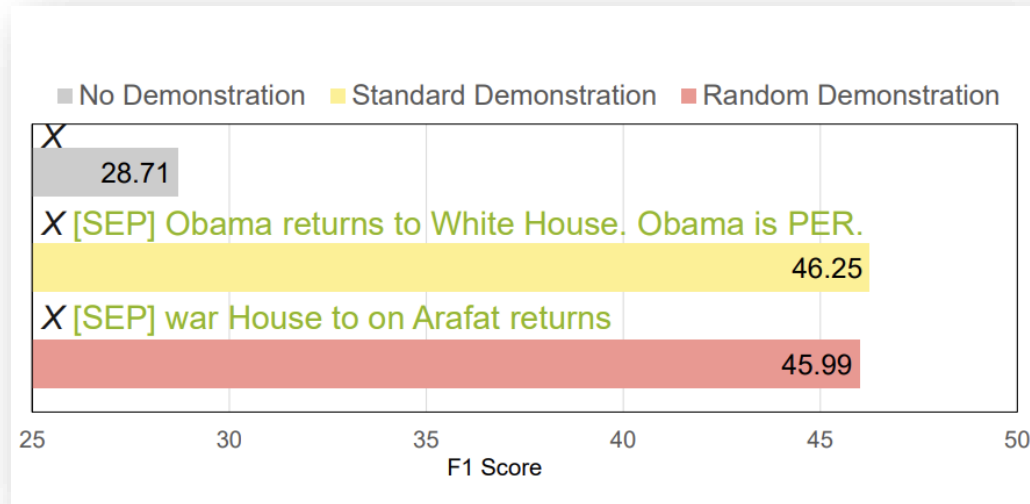
✓ **Prompting** (20 mins)

- ✓ Zero-shot, few-shot

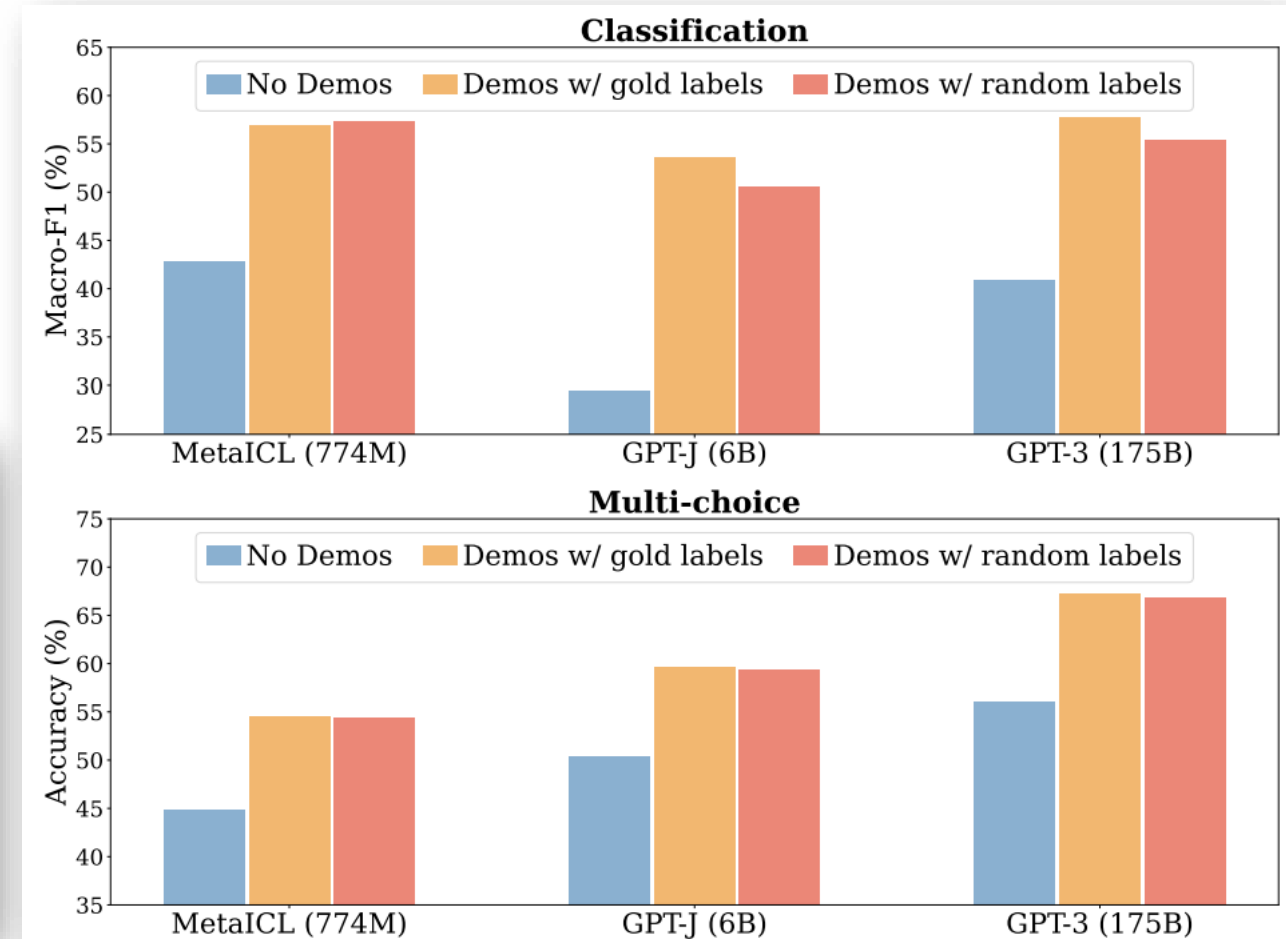
- ✓ Chain-of-thought

- Sensitivity and inconsistency

Random Labels Perform Similarly to Gold Labels



Random demonstrations in NER
([Zhang et al., 2022](#))



Random demonstrations in classification and multiple-choices
([Min et al., 2022](#))

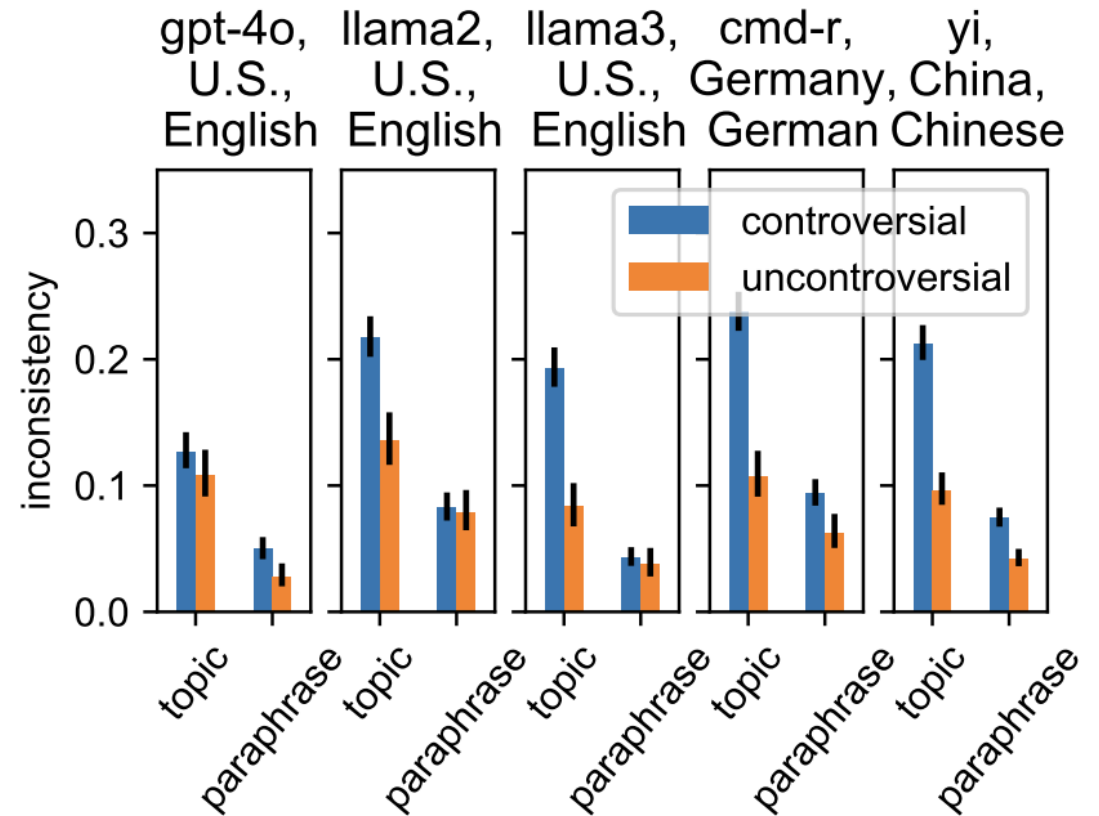
Inconsistency in Controversial Questions

Initial Question: Do you think that euthanasia is morally acceptable?

Topic: Do you think individuals should have the right to decide how they die?

Paraphrase: In your view, is euthanasia morally acceptable?

Multilingual: 你认为安乐死在道德上可接受吗?



Are Large Language Models Consistent over Value-laden Questions? Jared Moore, Tanvi Deshpande, Diyi Yang. arXiv:2407.02996

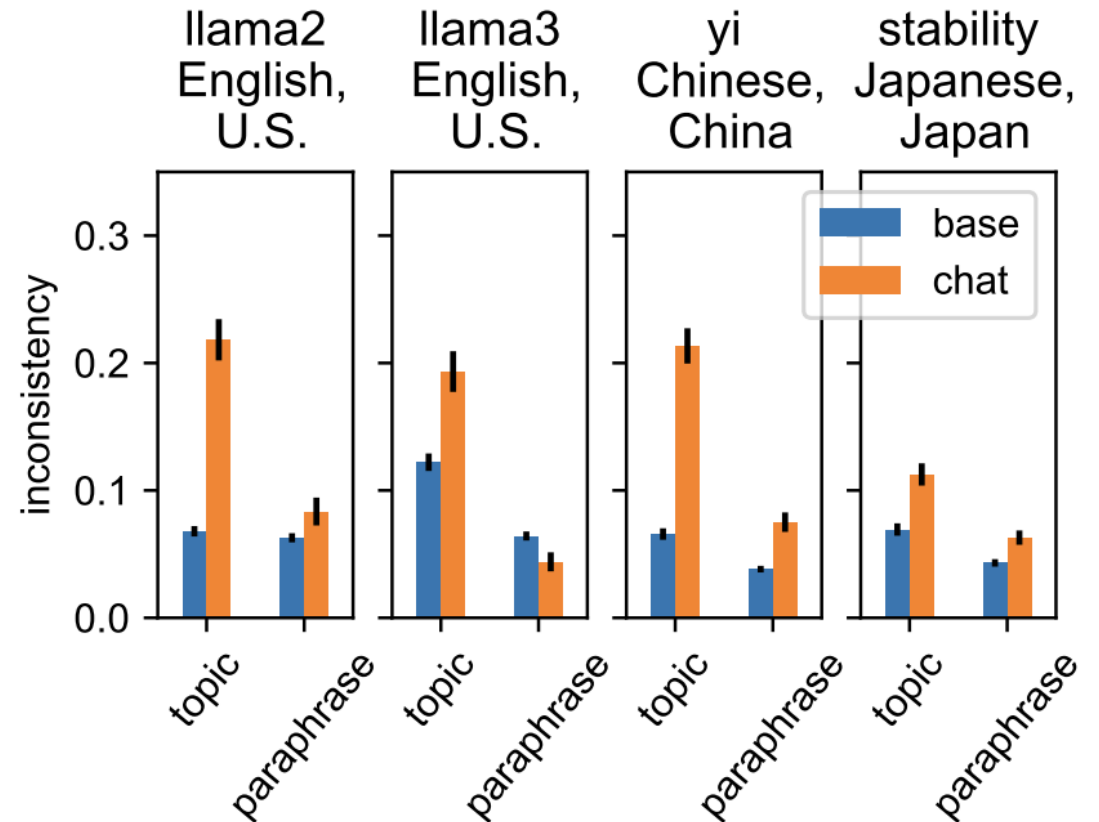
Inconsistency in Controversial Questions

Initial Question: Do you think that euthanasia is morally acceptable?

Topic: Do you think individuals should have the right to decide how they die?

Paraphrase: In your view, is euthanasia morally acceptable?

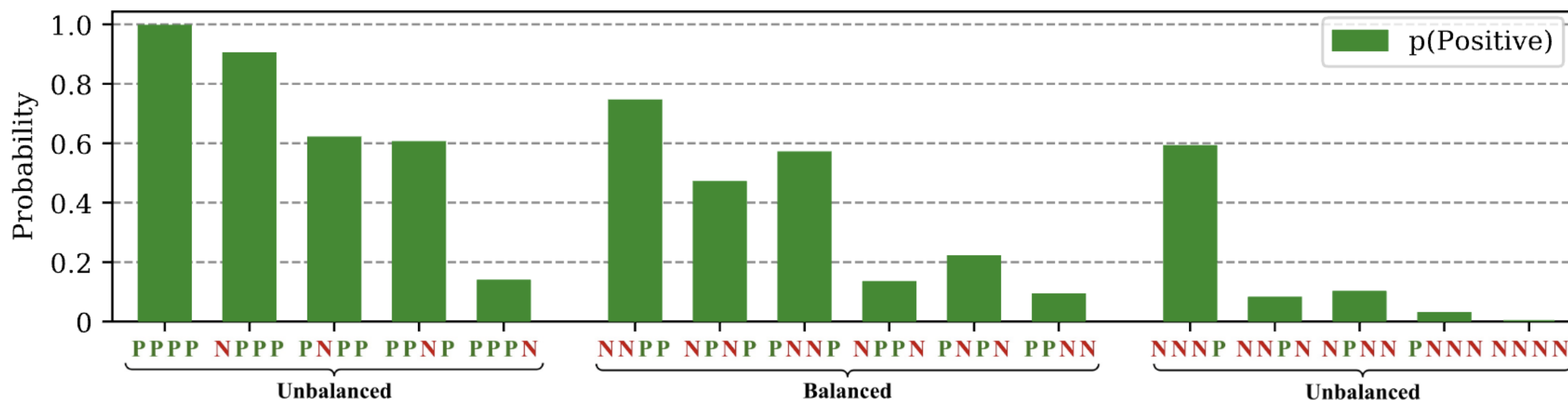
Multilingual: 你认为安乐死在道德上可接受吗?



Are Large Language Models Consistent over Value-laden Questions? Jared Moore, Tanvi Deshpande, Diyi Yang. arXiv:2407.02996

Majority Label and Recency Biases

Majority label and recency biases cause GPT-3 to become biased towards certain answers and help to explain the high variance across examples & orderings



First Token vs. Text Output

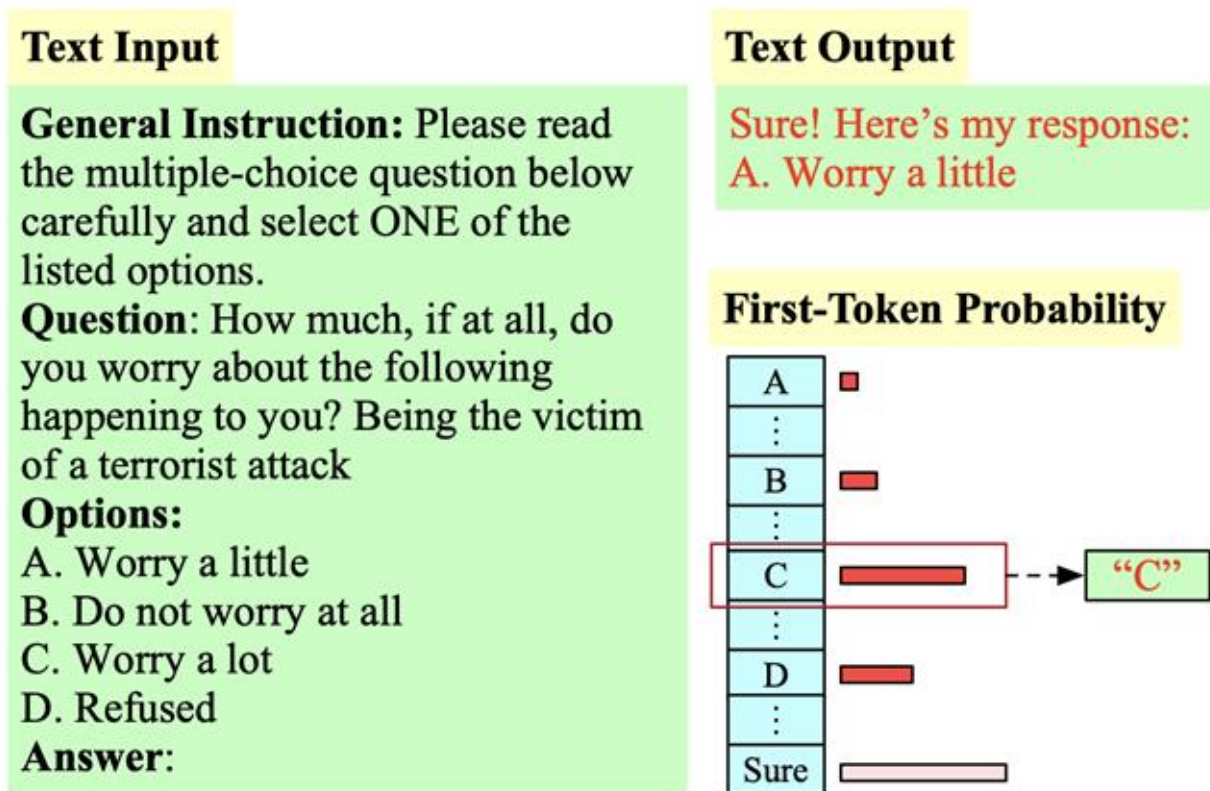


Figure 1: Example of LLM’s *mismatch* between first-token probability prediction (“C”) and text output (“A”).

Model (0-shot)	First Token	Text Answer
Gemma-7b-Inst	30.2	50.8
Llama2-7b-Chat	34.9	43.1
Llama2-13b-Chat	40.2	47.6
Mistral-7b-Inst-0.2	53.2	53.6

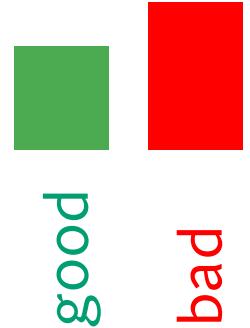
Table 1: Performance of the models on MMLU with different evaluation methods: First token probability and Text answer. Text answer achieves better performance in general.

Prior versus Context in Prompting

Context

Query

Trump hugs Biden. The relationship between Trump and Biden is...

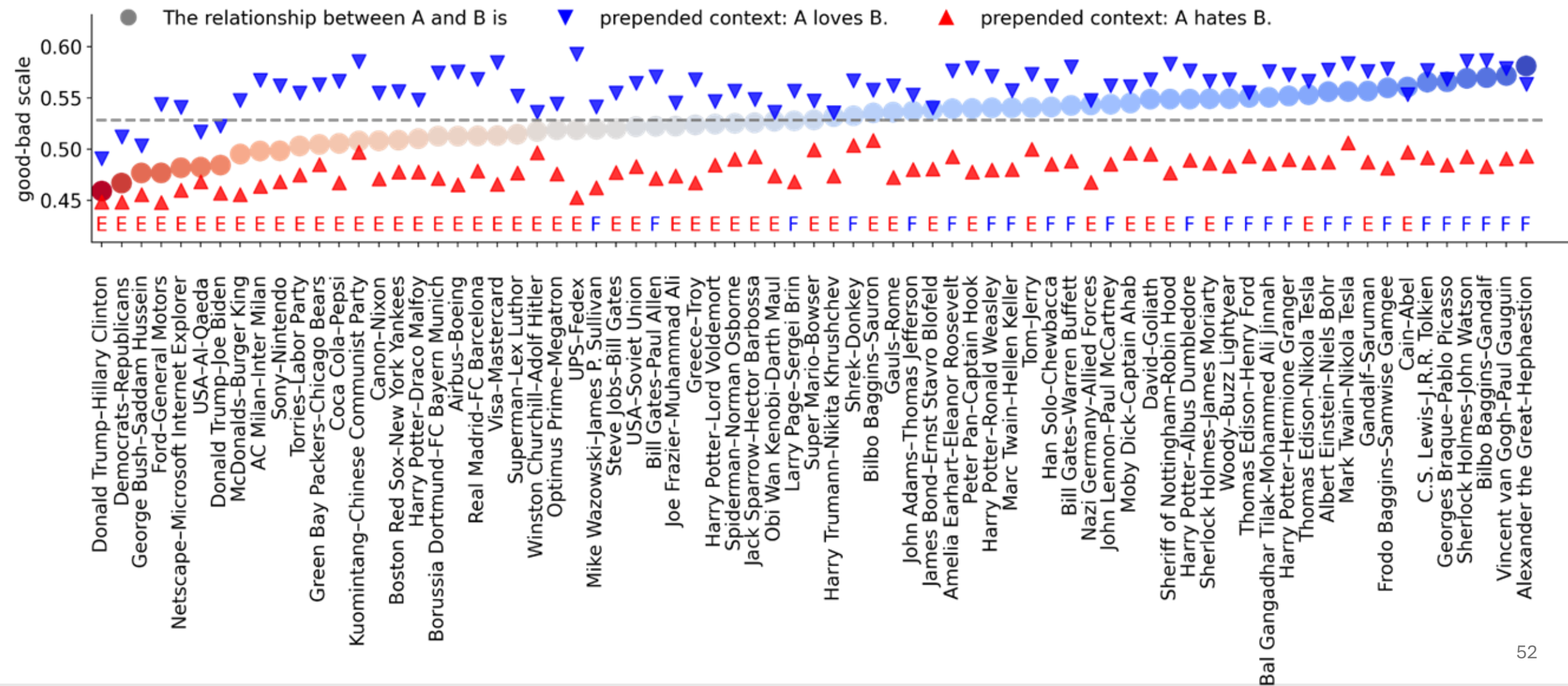


Likelihood

Prior



Prior versus Context in Prompting



Summary of Prompting Sensitivity

- Random labels
- Inconsistency
- Majority labels and recency bias
- Ordering effect
- First token vs. text output
- Prior vs. context in prompting
- ...



Outline

- ✓ **Transformers and Large Language Models** (30 mins)
- ✓ **Prompting** (20 mins)
 - ✓ Zero-shot, few-shot, chain-of-thought
 - ✓ Sensitivity and inconsistency
- ✓ **Optimization and Calibration** (20mins)

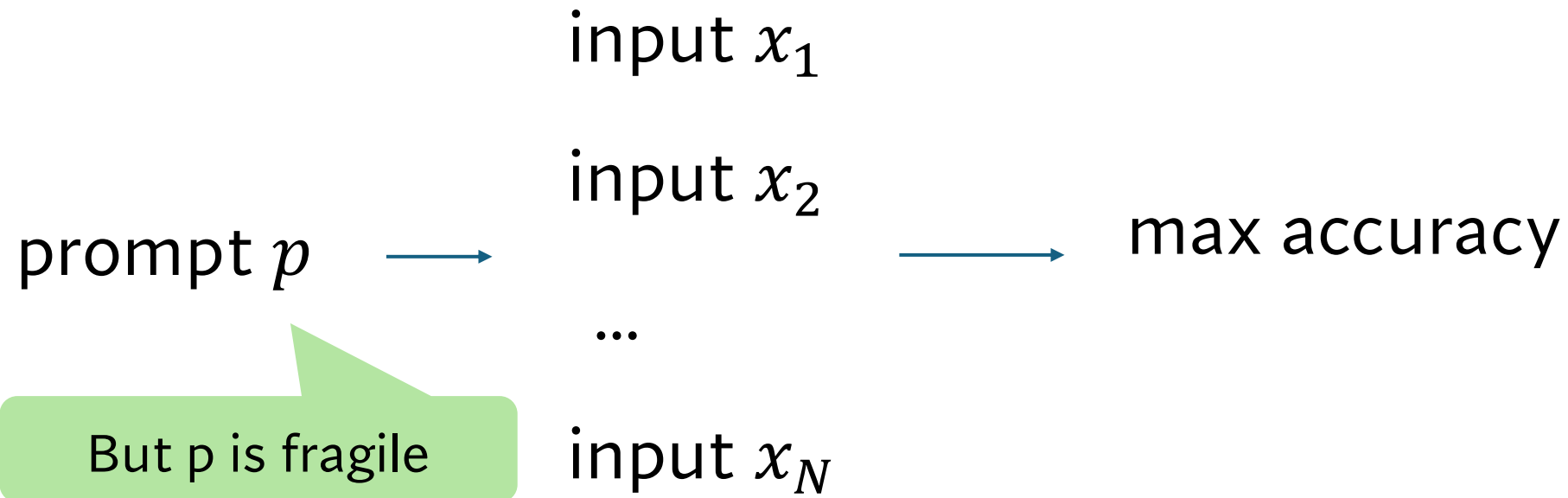
Downside of prompt-based learning

1. **Inefficiency:** The prompt needs to be processed *every time* the model makes a prediction.
2. **Poor performance vs. fine-tuning:** Prompting generally performs worse than fine-tuning [[Brown et al., 2020](#)].
3. **Sensitivity** to the wording of the prompt [[Webson & Pavlick, 2022](#)], order of examples [[Zhao et al., 2021](#); [Lu et al., 2022](#)], etc.
4. **Lack of clarity** regarding what the model learns from the prompt. Even random labels work [[Zhang et al., 2022](#); [Min et al., 2022](#)]!

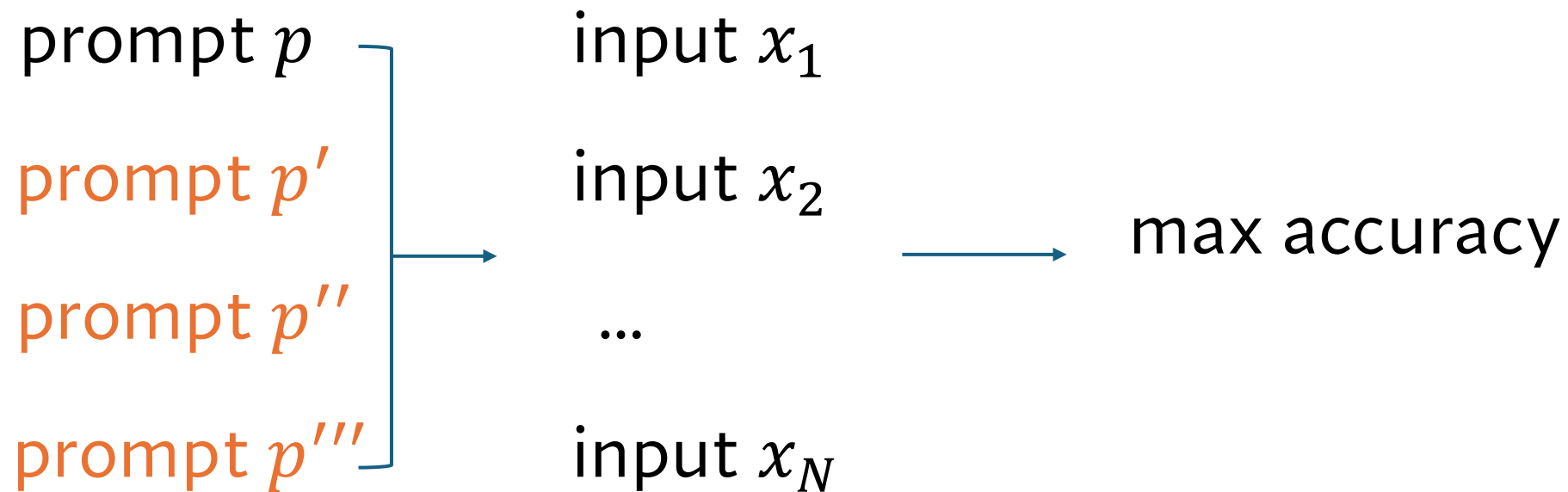
Status in 2026

1. **Inefficiency:** Still true, but mitigated. We now have better infrastructure, including advanced **prompt/KV caching** strategies.
2. **Poor performance vs. fine-tuning:** No longer generally true.
 - Prompting strong models can be extremely competitive
 - Fine-tuning is useful when you need consistent specialized behavior, or behavior that is hard to articulate.
3. **Sensitivity:** Still true, but mitigated.
4. **Lack of clarity** regarding what the model learns from the prompt: Still an open scientific questions

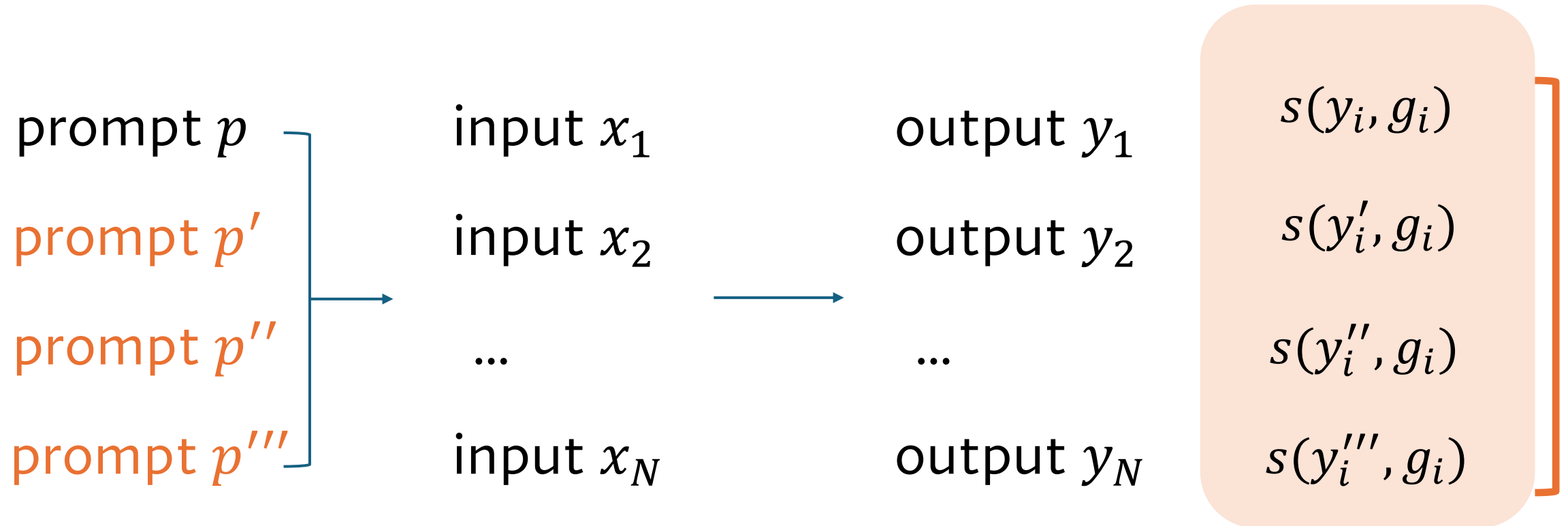
Prompt Optimization: *example on spam detection* $\{0,1\}$



Prompt Optimization: *example on spam detection* $\{0,1\}$



Prompt Optimization: *example on spam detection* $\{0,1\}$



How to find p ? Searching over all string space is intractable!

Find the prompt p that leads to the least error

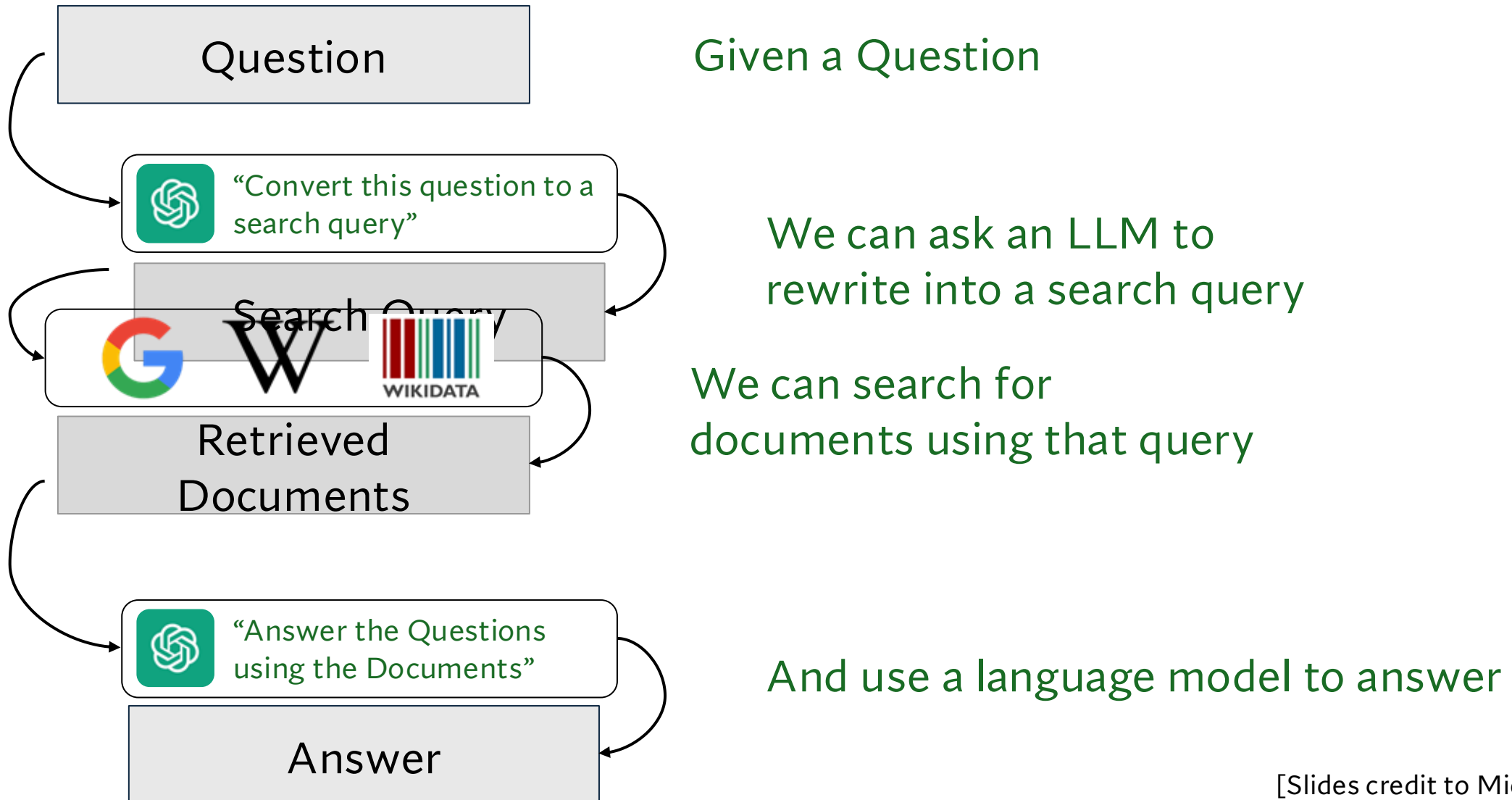
Programming vs. Not Prompting LLMs

Using LMs to *build & optimize* modular programs

that are more accurate, transparent, & efficient than LMs in isolation.



Example: using prompting for RAG



Instead of Tweaking Brittle Prompts...

Solve a question answering task with interleaving Thought, Action, Observation steps. Thought can reason about the current situation, and Action can be three types:

- (1) Search[entity], which searches the exact entity on Wikipedia and returns the first paragraph if it exists. If not, it will return some similar entities to search.
- (2) Lookup[keyword], which returns the next sentence containing keyword in the current passage.
- (3) Finish[answer], which returns the answer and finishes the task.

Here are some examples.

Question: What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?

Thought 1: I need to search Colorado orogeny, find the area that the eastern sector of the Colorado orogeny extends into, and find the elevation range of the area.

Action 1: Search[Colorado orogeny]

Observation 1: The Colorado orogeny was an episode of mountain building (an orogeny) in Colorado and surrounding areas.

Thought 2: It does not mention the eastern sector. So I need to look up eastern sector.

Action 2: Lookup[eastern sector]

Observation 2: (Result 1 / 1) The eastern sector extends into the High Plains and is called the Central Plains orogeny.

Thought 3: The eastern sector of Colorado orogeny extends into the High Plains. So I need to search High Plains.

Action 3: Search[High Plains]

[... truncated ...]

Scores

33%

with **GPT-3.5**
on a multi-hop
QA task

Write a short LM program in Python

... and let the program learn how to use the LM effectively!

```
class MultiHop(dspy.Module):
    def __init__(self):
        self.generate_query = dspy.ChainOfThought("context, question -> query")
        self.generate_answer = dspy.ChainOfThought("context, question -> answer")

    def forward(self, question):
        context = []
        for hop in range(2):
            query = self.generate_query(context, question).query
            context += dspy.Retrieve(k=3)(query).passages
        answer = self.generate_answer(context, question)

        return answer
```

```
self.generate_query = dsp.ChainOfThought("context, question -> query")
```

Carefully read the provided `context` and `question`. Your task is to formulate a concise and relevant `query` that could be used to retrieve information from a search engine to answer the question most effectively. The `query` should encapsulate...



Automatically Generated Instruction

```
self.generate_query = dsp.ChainOfThought("context, question -> query")
```

Carefully read the provided `context` and `question`. Your task is to formulate a concise and relevant `query` that could be used to retrieve information from a search engine to answer the question most effectively. The `query` should encapsulate...

Context: [1] Twilight is a series of four vampire-themed fantasy romance...

[2] The Harper Connelly Mysteries is a series of fantasy...

Question: In which year was the first of the vampire-themed fantasy romance novels, for which The Twilight Saga serves as a spin-off encyclopedic reference book, first published?

Reasoning: Let's determine when that fantasy romance novel was first published.

Search Query: When was the first of the vampire-themed fantasy romance novels published?



Automatically Generated Example

```
self.generate_query = dsp.ChainOfThought("context, question -> query")
```

Carefully read the provided `context` and `question`. Your task is to formulate a concise and relevant `query` that could be used to retrieve information from a search engine to answer the question most effectively. The `query` should encapsulate...

Context: [1] Twilight is a series of four vampire-themed fantasy romance...
[2] The Harper Connelly Mysteries is a series of fantasy...

Question: In which year was the first of the vampire-themed fantasy roma...
which The Twilight Saga serves as a spin-off encyclopedic f

Reasoning: Let's determine when that fantasy romance nove... is...
Search Query: When was the first of the vampire-themed fa... vels published?

Context: [1] The Victorians - Their Story In Pictures is...
[2] The Caxtons: A Family Picture is an 1849 Vic

Question: The Victorians is a documentary series written

Reasoning: We know that the documentary series is about Victorian art and...
was written by Jeremy Paxman. We need to find the year in which Jeremy Paxm

Search Query: Jeremy Paxman birth year

Scores

55%

with GPT-3.5
on multi-hop QA

39%

with T5-770M
(+ finetuning)

50%

with Llama2-13B

* prompt parts adapted & con

Tune Prompts and LM Weights in Arbitrary Programs

1. **Bootstrap demonstrations**
2. **Update the program** with them
 - Automatic few-shot prompting
 - Induction of revised instructions: *dspy.MIPROv2*
 - Multi-stage fine-tuning: *dspy.BootstrapFinetune*
3. **Compose optimizers** into complex meta-pipelines



Prompt Optimization tutorial:
<https://dspy.ai/tutorials/math/>

Summary

✓ **Transformers and Large Language Models**

✓ **Prompting**

- ✓ Zero-shot, few-shot, chain-of-thought
- ✓ Sensitivity and inconsistency

✓ **Optimization and Calibration**

WE DID IT



Next Class: Learning from human preferences