

CS11-711 Advanced NLP

# Prompting and Instruction Tuning

Daniel Fried



**Carnegie Mellon University**  
Language Technologies Institute

Site

[cmu-anlp.github.io](https://cmu-anlp.github.io)

With slides from Greg Durrett, Pengfei Liu, and Graham Neubig

# Five Paradigms of NLP Progress

- Feature Engineering
- Architecture Engineering
- Objective Engineering
- Prompt Engineering
- Multi-task Model Engineering

# Feature Engineering

- **Paradigm:** Fully Supervised Learning (Non-neural Network)
- **Time Period:** Most popular through 2015
- **Characteristics:**
  - Non-neural machine learning models mainly used
  - Require manually defined feature extraction
- **Representative Work:**
  - Manual features -> linear or multinomial regression
  - Manual features -> conditional random fields (CRF)

# Architecture Engineering

- **Paradigm:** Fully Supervised Learning (Neural Networks)
- **Time Period:** About 2013-2018
- **Characteristics:**
  - Rely on neural networks
  - Do not need to manually define features, but should modify the network structure (e.g.: LSTM v.s CNN)
  - Sometimes used pre-training of LMs, but often only for shallow features such as embeddings
- **Representative Work:**
  - LSTMs for Text Classification

# Objective Engineering

- **Paradigm:** Pre-train, Fine-tune
- **Time Period:** 2017-Now
- **Characteristics:**
  - Pre-trained LMs (PLMs) used as initialization of full model - both shallow and deep features
  - Less work on architecture design, but engineer objective functions
- **Typical Work:**
  - BERT, BART, T5

# Prompt Engineering

- **Paradigm:** Pre-train, Prompt, Predict
- **Date:** 2019-Now
- **Characteristic:**
  - NLP tasks are modeled entirely by relying on LMs
  - The tasks of shallow and deep feature extraction, and prediction of the data are all given to the LM
  - Engineering of prompts is required
- **Representative Work:**
  - GPT3

# Multi-Task Model Engineering

- **Paradigm:** Pre-train, Instruction-tuning/RL from human feedback, prompt.
- **Date:** 2021-Now
- **Characteristic:**
  - Train an LLM to be able to do multiple tasks, conditioned on a task description
  - Doesn't require fine-tuning
  - Makes the model more robust to prompts
  - Requires rich data from multiple tasks
- **Representative Work:**
  - T0, FLAN, InstructGPT

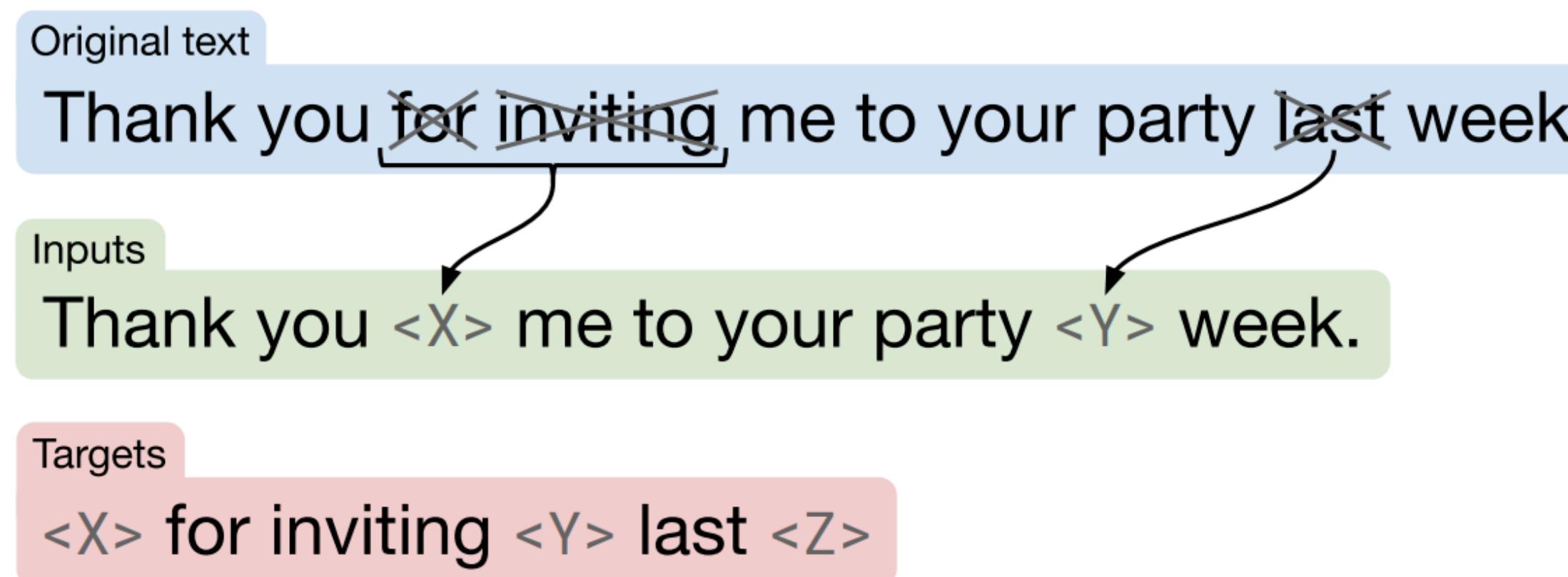
# **Objective Engineering:**

## **T5 Recap**

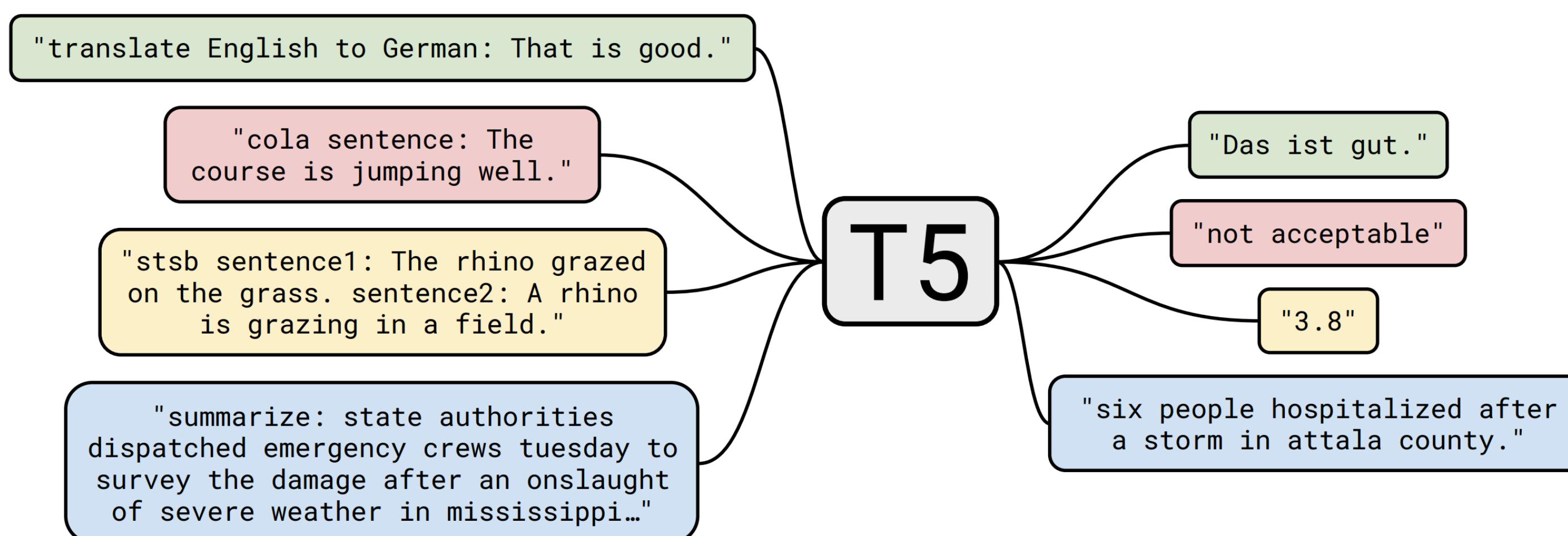
# T5

De-noising seq2seq pre-training allows many tasks in fine-tuning

Pre-training:



Fine-tuning:



# UnifiedQA

Multiple choice

MC	<b>Dataset</b>	MCTest
	<b>Input</b>	Who was Billy? \n (A) The skinny kid (B) A teacher (C) A little kid (D) The big kid \n Billy was like a king on the school yard. A king without a queen. He was the biggest kid in our grade, so he made all the rules during recess. ...
YN	<b>Output</b>	The big kid
	<b>Dataset</b>	BoolQ
YN	<b>Input</b>	Was America the first country to have a president? \n (President) The first usage of the word president to denote the highest official in a government was during the Commonwealth of England ...
	<b>Output</b>	no

- ▶ Past work: different architectures for every QA formulation. (Span selection, answer generation, multiple choice, ...)
- ▶ Now: one 11B parameter T5 model

# UnifiedQA

<b>Dataset</b>	NarrativeQA
<b>Input</b>	What does a drink from narcissus's spring cause the drinker to do? \n Mercury has awakened Echo, who weeps for Narcissus, and states that a drink from Narcissus's spring causes the drinkers to ``Grow dotingly enamored of themselves.'' ...
<b>Output</b>	fall in love with themselves

Abstractive question, requires generating *free-form answer*

- ▶ Past work: different architectures for every QA formulation. (Span selection, answer generation, multiple choice, ...)
- ▶ Now: one 11B parameter T5 model

# UnifiedQA

- ▶ Model gets decent results on held-out datasets
- ▶ But, lots of the answers were likely seen in the T5 pre-training dataset!

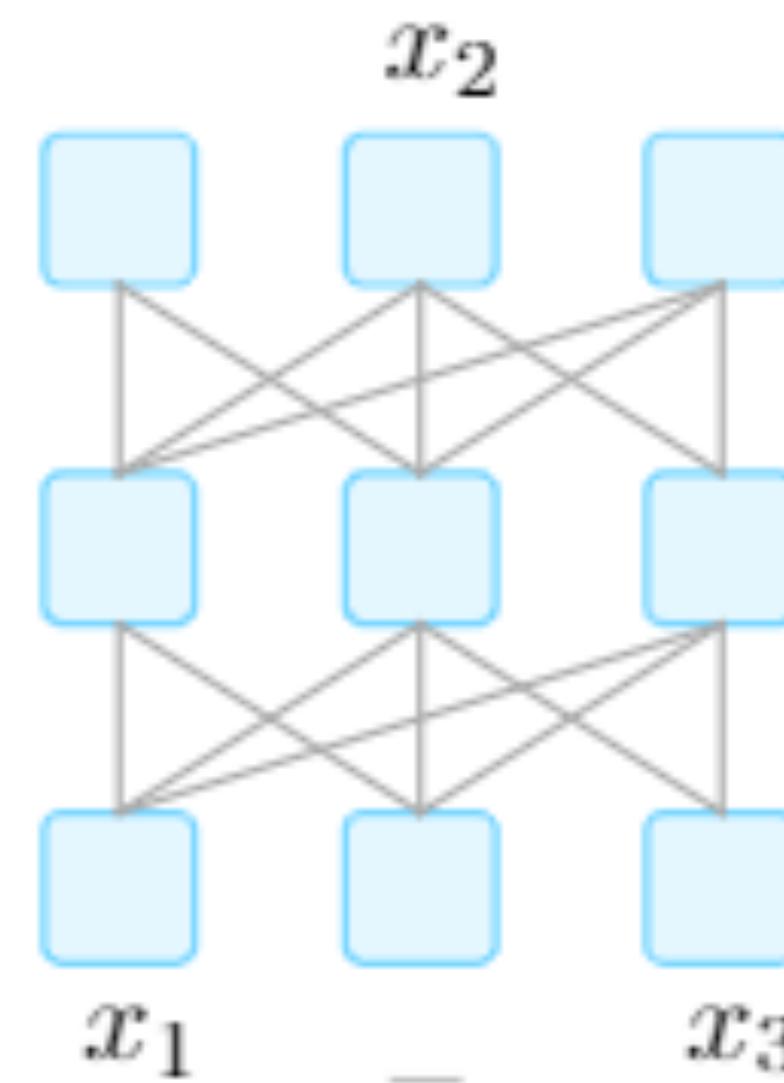
Seen dataset?	Model ↓ - Evaluated on →	NewsQA	Quoref	Quoref-CS	ROPES	ROPES-CS	DROP	DROP-CS	QASC	Common senseQA	NP-BoolQ	BoolQ-CS	MultiRC	Avg
No	UnifiedQA [EX]	58.7	64.7	53.3	43.4	29.4	24.6	24.2	55.3	62.8	20.6	12.8	7.2	38.1
	UnifiedQA [AB]	58.0	<b>68.2</b>	57.6	48.1	41.7	30.7	36.8	54.1	59.0	27.2	39.9	28.4	45.8
	UnifiedQA [MC]	48.5	67.9	<b>58.0</b>	61.0	44.4	28.9	37.2	67.9	75.9	2.6	5.7	9.7	42.3
	UnifiedQA [YN]	0.6	1.7	1.4	0.0	0.7	0.4	0.1	14.8	20.8	79.1	78.6	<b>91.7</b>	24.2
	UnifiedQA	<b>58.9</b>	63.5	55.3	<b>67.0</b>	<b>45.5</b>	<b>32.5</b>	<b>40.1</b>	<b>68.5</b>	<b>76.2</b>	<b>81.3</b>	<b>80.4</b>	59.9	<b>60.7</b>
Yes	Previous best	66.8	86.1	55.4	61.1	32.5	89.1	54.2	85.2	79.1	78.4	71.1	--	
		Retro Reader	TASE	XLNet	ROBERTa	RoBERTa	ALBERT	MTMSN	KF+SIR+2Step	reeLB-RoBERT	RoBERTa	RoBERTa	--	

# Towards Multi-Task Models

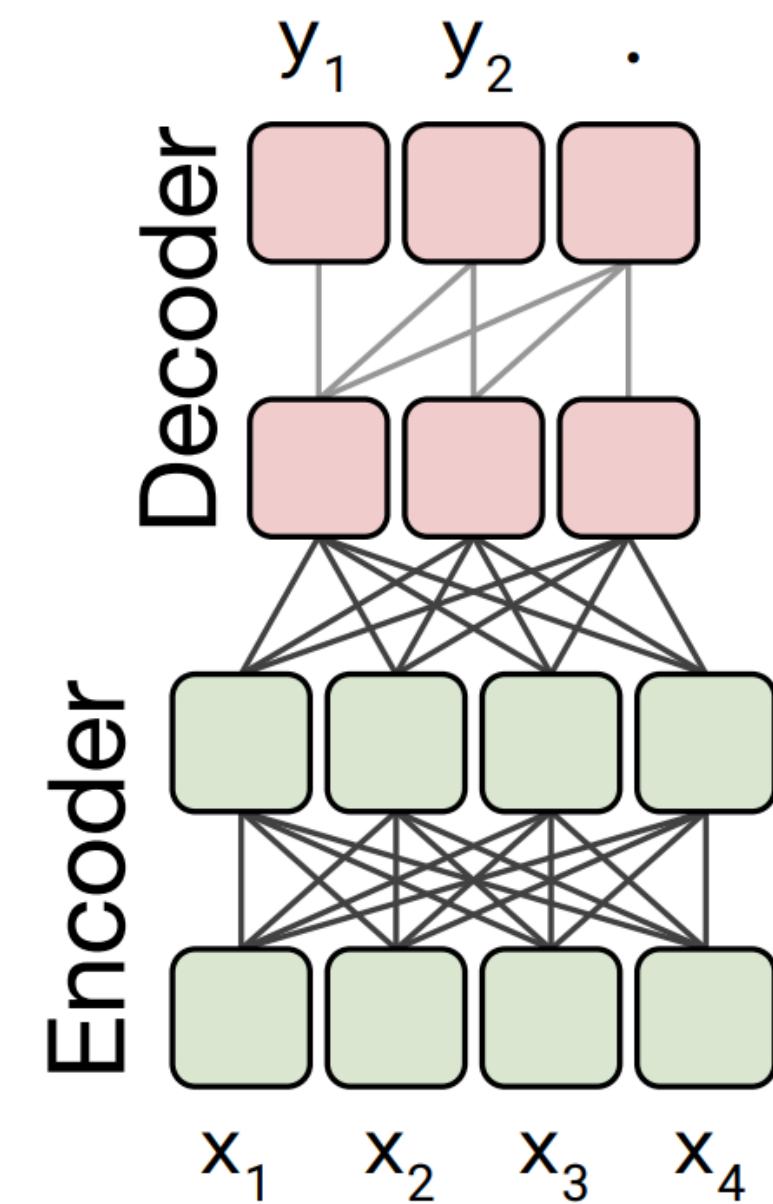
- BART and T5 are useful for all sorts of seq2seq tasks involving language — use these for seq-to-seq pre-training.  
(Caveat: need specialized models for language-to-code, like PLBART and CodeT5)
- UnifiedQA suggests that big generative models are good at generalizing across tasks and even to new tasks (although QA results have a long way to go)
- If we have a strong enough pre-trained model and train on enough tasks, can we generalize to new tasks?
- How do we specify those new tasks if they're not close to tasks we've already run on?
- Answer: prompting. But to do that well, we'll need to scale up further

# **Prompt Engineering: LLMs and GPT**

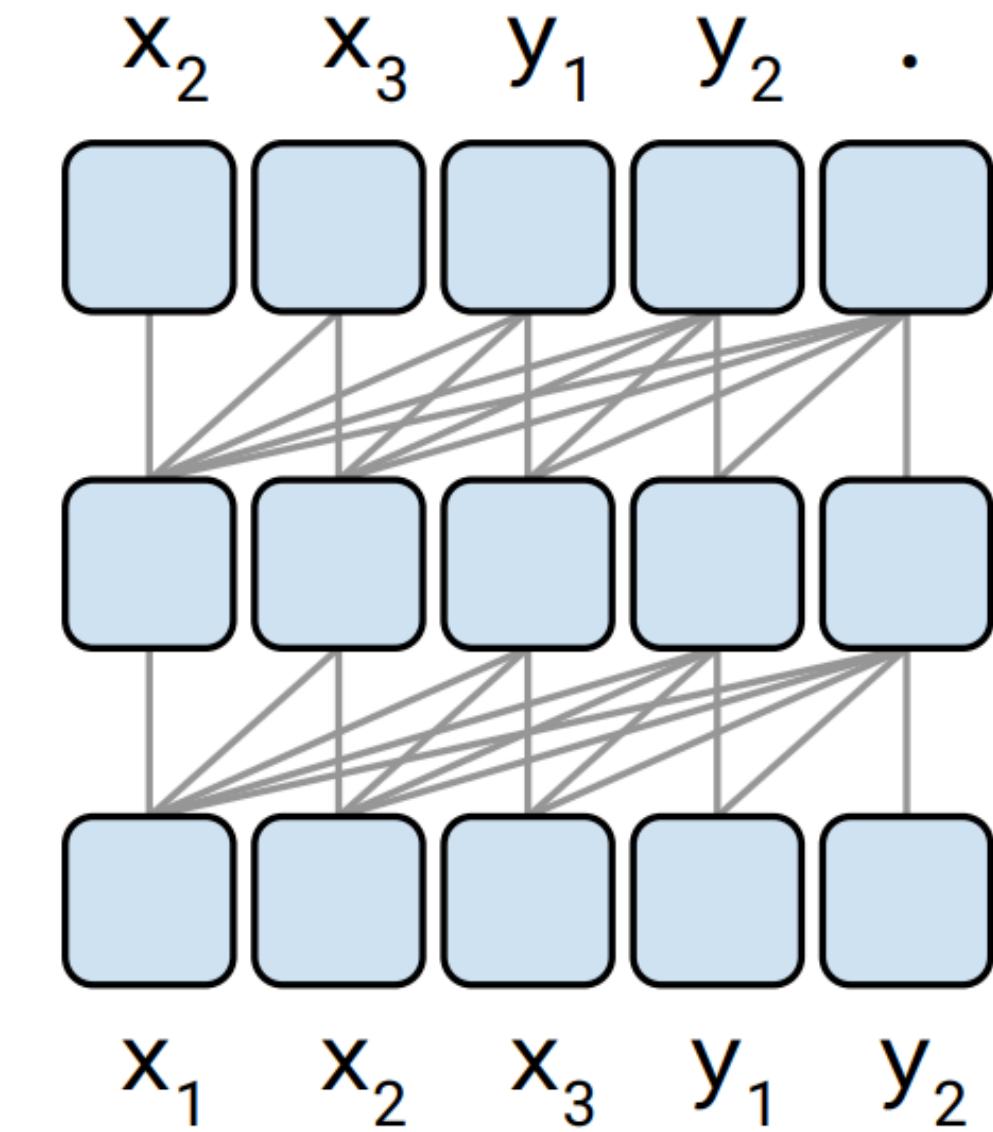
# Model Zoo



Masked Language Model  
(e.g. BERT)



Encoder-Decoder  
(e.g. BART, T5)



(Causal) Language Model  
(e.g. GPT)

# OpenAI GPT/GPT2

- ▶ Very large language models using the Transformer architecture
- ▶ Straightforward left-to-right language model, trained on raw text
- ▶ GPT2: trained on 40GB of text

	Parameters	Layers	$d_{model}$
approximate size of BERT	117M	12	768
	345M	24	1024
	762M	36	1280
GPT-2	1542M	48	1600

- ▶ By far the largest of these models trained when it came out in March 2019
- ▶ Because it's a language model, we can **generate** from it

# OpenAI GPT2

**SYSTEM PROMPT  
(HUMAN-WRITTEN)**

*Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today.*

**MODEL COMPLETION  
(MACHINE-WRITTEN,  
SECOND TRY)**

The 19-year-old singer was caught on camera being escorted out of the store by security guards.

The singer was wearing a black hoodie with the label ‘Blurred Lines’ on the front and ‘Fashion Police’ on the back.

Scroll down for video

Shoplifting: Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today (pictured)

slide credit: OpenAI

# Pre-Training Cost (with Google/AWS)

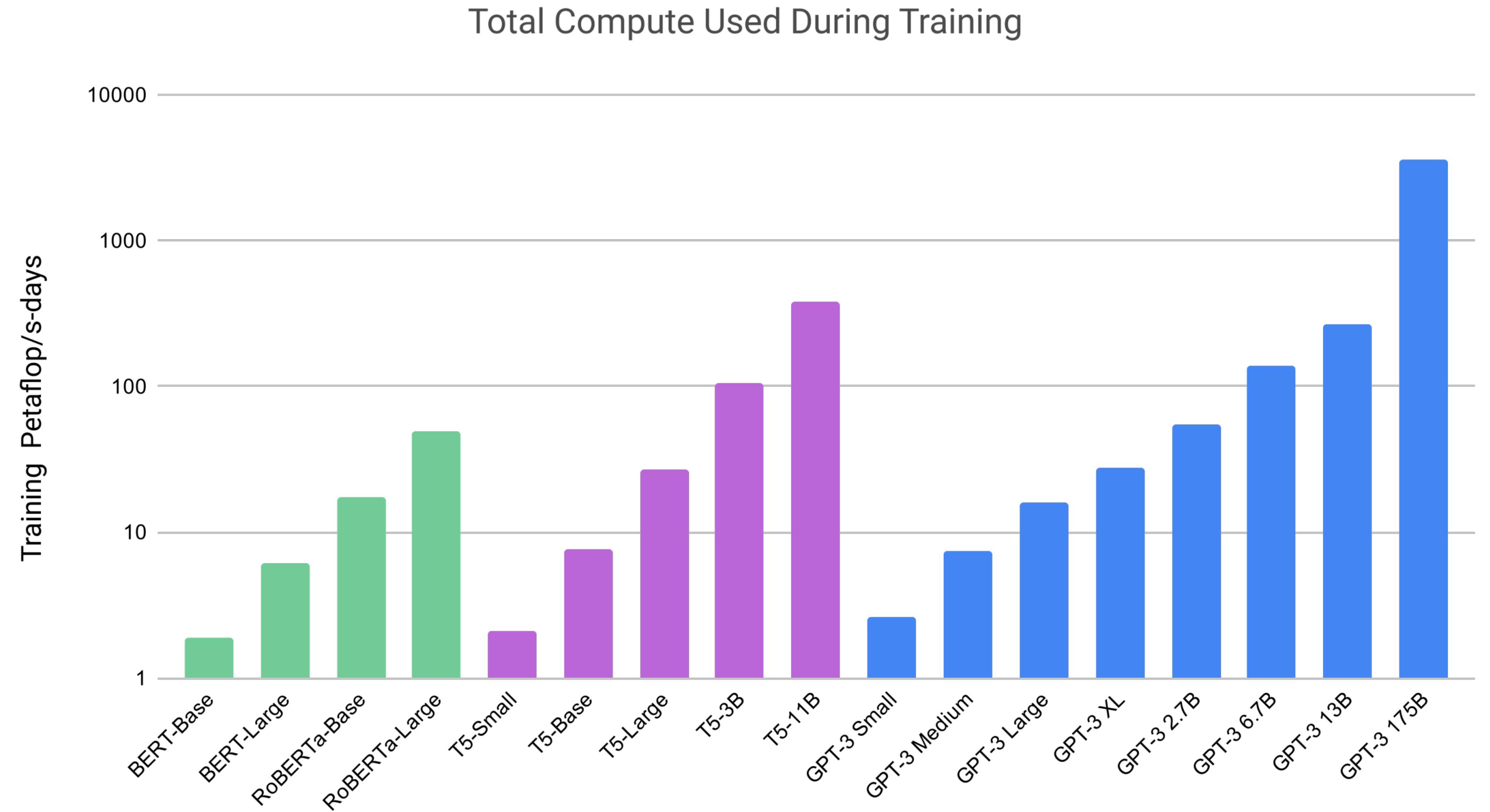
- ▶ BERT: Base \$500, Large \$7000
- ▶ GPT-2 (as reported in other work): \$25,000
- ▶ This is for a single pre-training run...developing new pre-training techniques may require many runs
- ▶ *Fine-tuning* these models can typically be done with a single GPU (but may take 1-3 days for medium-sized datasets)

# Prompts

- ▶ Prompts can help induce the model to engage in certain behavior
- ▶ In the GPT-2 paper, “tl;dr:” (too long; didn't read) is mentioned as a prompt that frequently shows up in the wild **indicating a summary**
- ▶ tl;dr is an indicator that the model should “switch into summary mode” now — and if there are enough clean instances of tl;dr in the wild, maybe the model has been trained on a ton of diverse data?

# Pushing the Limits: GPT-3

- ▶ 175B parameter model: 96 layers, 96 heads, 12k-dim vectors
- ▶ Trained on Microsoft Azure, estimated to cost roughly \$10M



# Where's the Computation?

1	description	FLOPs / update	% FLOPS MHA	% FLOPS FFN	% FLOPS attn	% FLOPS logit
8	OPT setups					
9	760M	4.3E+15	35%	44%	14.8%	5.8%
10	1.3B	1.3E+16	32%	51%	12.7%	5.0%
11	2.7B	2.5E+16	29%	56%	11.2%	3.3%
12	6.7B	1.1E+17	24%	65%	8.1%	2.4%
13	13B	4.1E+17	22%	69%	6.9%	1.6%
14	30B	9.0E+17	20%	74%	5.3%	1.0%
15	66B	9.5E+17	18%	77%	4.3%	0.6%
16	175B	2.4E+18	17%	80%	3.3%	0.3%
..						

In big transformer models (e.g. GPT-3),  
most computation is in the multi-layer perceptrons

Credit: Stephen Roller

<https://twitter.com/stephenroller/status/1579993017234382849/photo/1>

# Pre-GPT-3: Fine-tuning

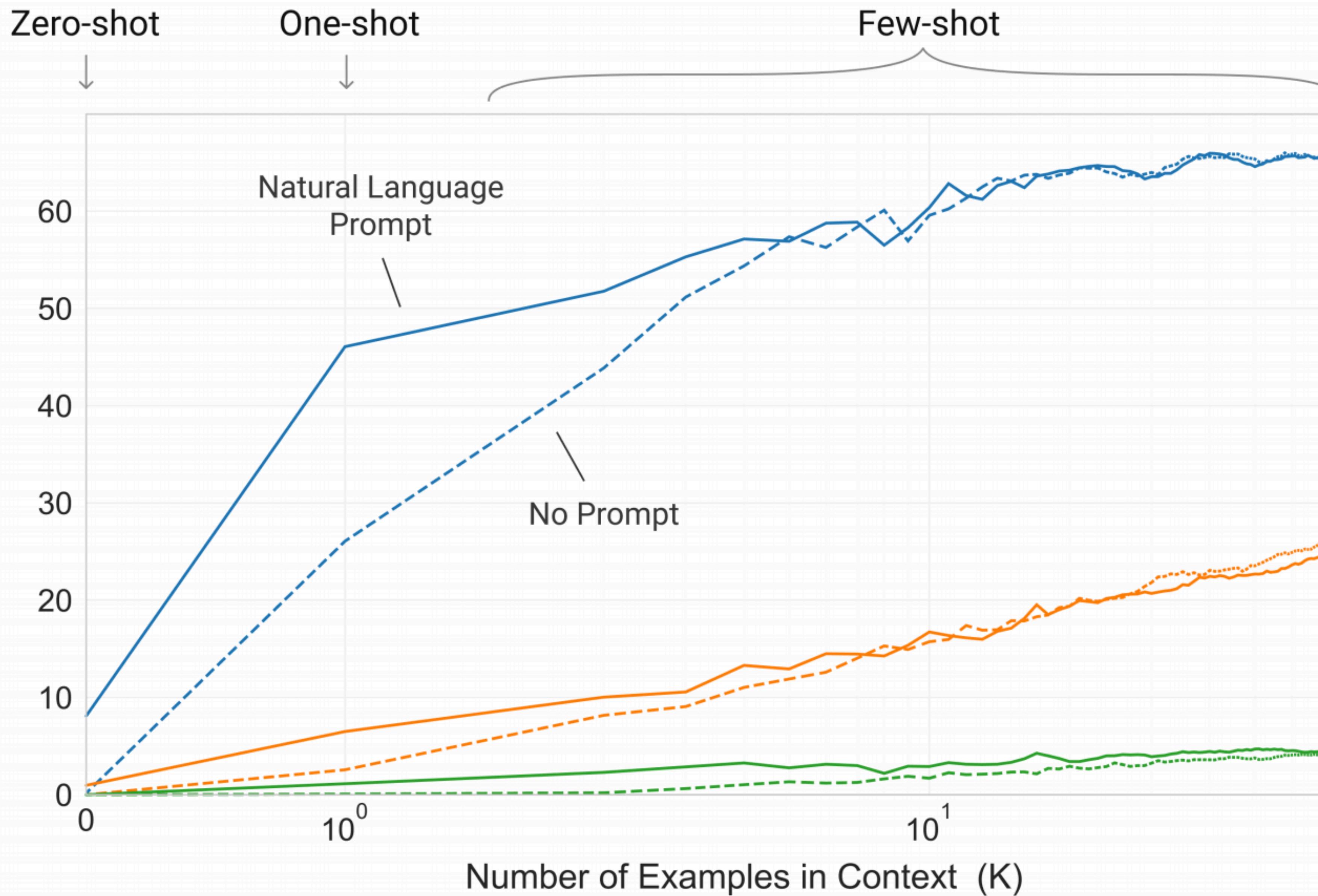
- ▶ Fine-tuning: this is the “normal way” of doing learning in models like GPT-2
- ▶ Requires computing the gradient and applying a parameter update on every example
- ▶ **This is super expensive with 175B parameters**



# GPT-3: Few-shot Learning

- ▶ GPT-3 proposes an alternative: **in-context learning**. Just uses the off-the-shelf model, no gradient updates
  - ▶ This procedure depends heavily on the examples you pick as well as the prompt (“*Translate English to French*”)
- 
- ```
graph LR; TD["1 Translate English to French:"] --> TD_desc["task description"]; E["2 sea otter => loutre de mer  
3 peppermint => menthe poivrée  
4 plush girafe => girafe peluche  
5 cheese => ....."] --> E_ex["examples"]; P["cheese => ....."] --> P_prom["prompt"]
```
- 1 Translate English to French: ← task description
- 2 sea otter => loutre de mer ← examples
- 3 peppermint => menthe poivrée ←
- 4 plush girafe => girafe peluche ←
- 5 cheese => ..... ← prompt

# GPT-3



**175B Params**

- **Key observation:** few-shot learning works best with huge models!

**13B Params**

**1.3B Params**

Brown et al. (2020)

# GPT-3

|                       | SuperGLUE<br>Average | BoolQ<br>Accuracy | CB<br>Accuracy | CB<br>F1    | COPA<br>Accuracy | RTE<br>Accuracy |
|-----------------------|----------------------|-------------------|----------------|-------------|------------------|-----------------|
| Fine-tuned SOTA       | <b>89.0</b>          | <b>91.0</b>       | <b>96.9</b>    | <b>93.9</b> | <b>94.8</b>      | <b>92.5</b>     |
| Fine-tuned BERT-Large | 69.0                 | 77.4              | 83.6           | 75.7        | 70.6             | 71.7            |
| GPT-3 Few-Shot        | 71.8                 | 76.4              | 75.6           | 52.0        | 92.0             | 69.0            |

|                       | WiC<br>Accuracy | WSC<br>Accuracy | MultiRC<br>Accuracy | MultiRC<br>F1a | ReCoRD<br>Accuracy | ReCoRD<br>F1 |
|-----------------------|-----------------|-----------------|---------------------|----------------|--------------------|--------------|
| Fine-tuned SOTA       | <b>76.1</b>     | <b>93.8</b>     | <b>62.3</b>         | <b>88.2</b>    | <b>92.5</b>        | <b>93.3</b>  |
| Fine-tuned BERT-Large | 69.6            | 64.6            | 24.1                | 70.0           | 71.3               | 72.0         |
| GPT-3 Few-Shot        | 49.4            | 80.1            | 30.5                | 75.4           | 90.2               | 91.1         |

- ▶ Sometimes very impressive, (MultiRC, ReCoRD), sometimes very bad
- ▶ Results on other datasets are equally mixed – but still strong for a few-shot model!

# Recommended Reading on Prompting

---

## Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing

---

**Pengfei Liu**

Carnegie Mellon University  
pliu3@cs.cmu.edu

**Weizhe Yuan**

Carnegie Mellon University  
weizhey@cs.cmu.edu

**Jinlan Fu**

National University of Singapore  
jinlanjonna@gmail.com

**Zhengbao Jiang**

Carnegie Mellon University  
zhengbaej@cs.cmu.edu

**Hiroaki Hayashi**

Carnegie Mellon University  
hiroakih@cs.cmu.edu

**Graham Neubig**

Carnegie Mellon University  
gneubig@cs.cmu.edu



# Zero-shot Prompting

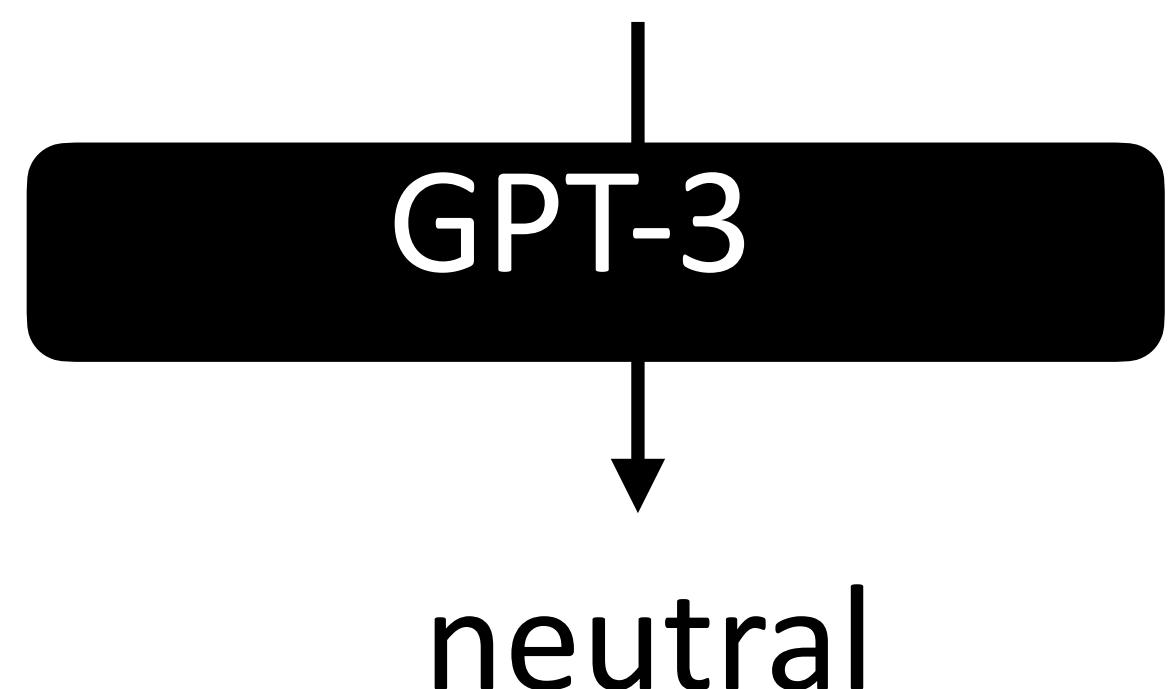
- ▶ Single unlabeled datapoint  $x$ , want to predict label  $y$

$x = \text{The movie's acting could've been better, but the visuals and directing were top-notch.}$

- ▶ Wrap  $x$  in a template we call a **verbalizer**  $v$

*Review: The movie's acting could've been better, but the visuals and directing were top-notch.*

*Out of positive, negative, or neutral, this review is*



# Zero-shot Prompting

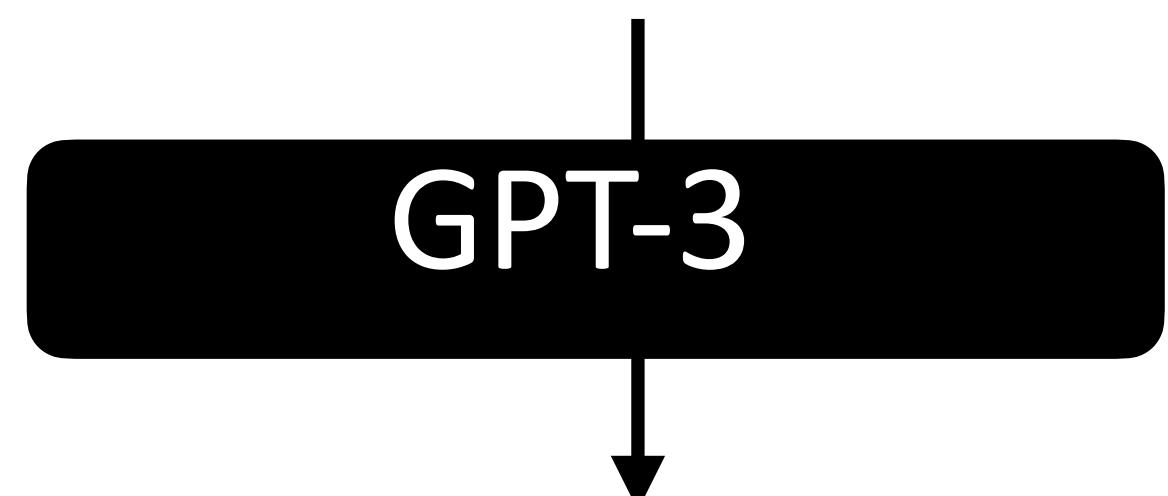
- ▶ Single unlabeled datapoint  $x$ , want to predict label  $y$

$x = \text{The movie's acting could've been better, but the visuals and directing were top-notch.}$

- ▶ Wrap  $x$  in a template we call a **verbalizer**  $v$

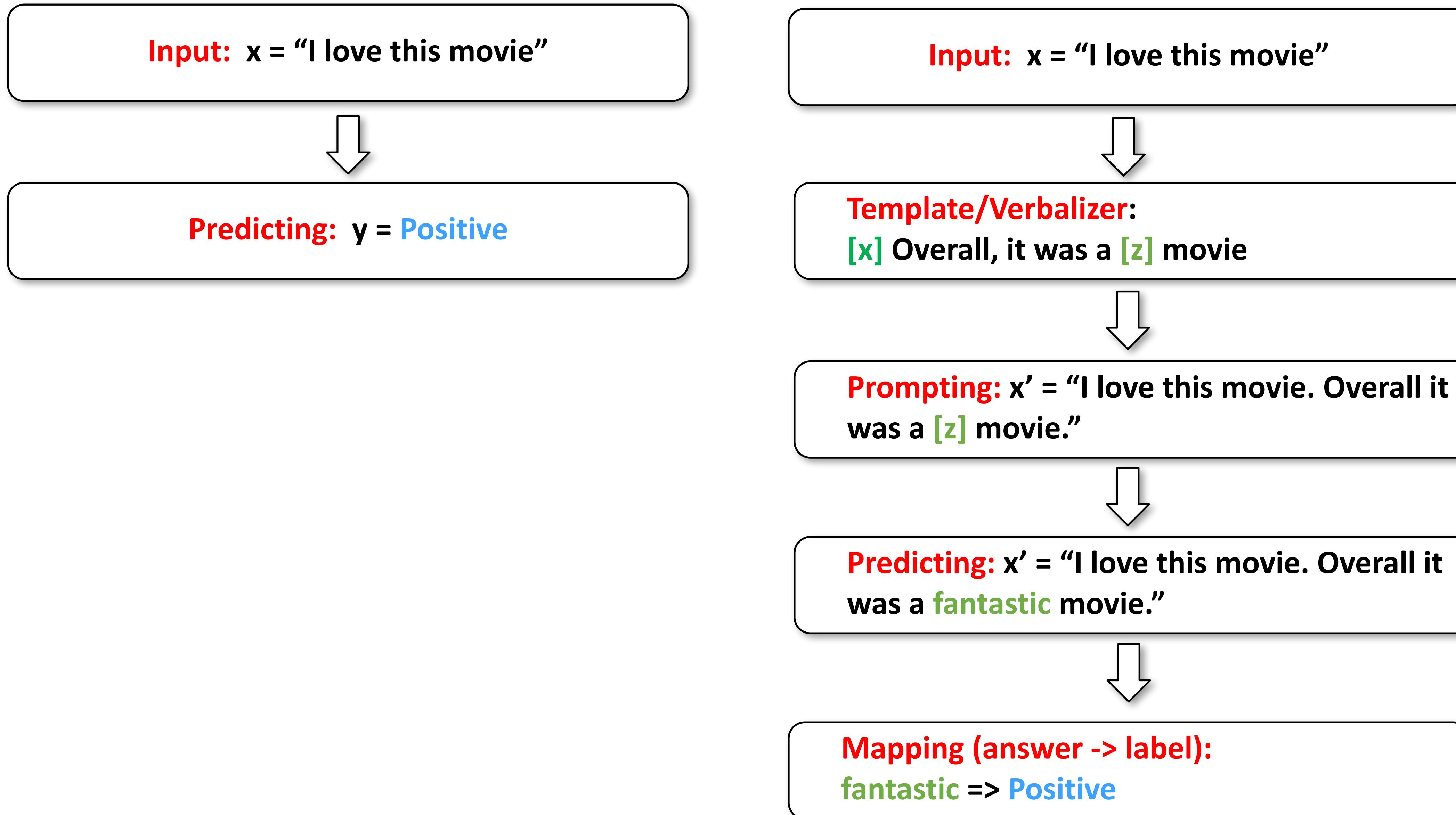
*Review: The movie's acting could've been better, but the visuals and directing were top-notch.*

*On a 1 to 4 star scale, the reviewer would probably give this movie*



3 stars.

# Traditional Formulation V.S Prompt Formulation

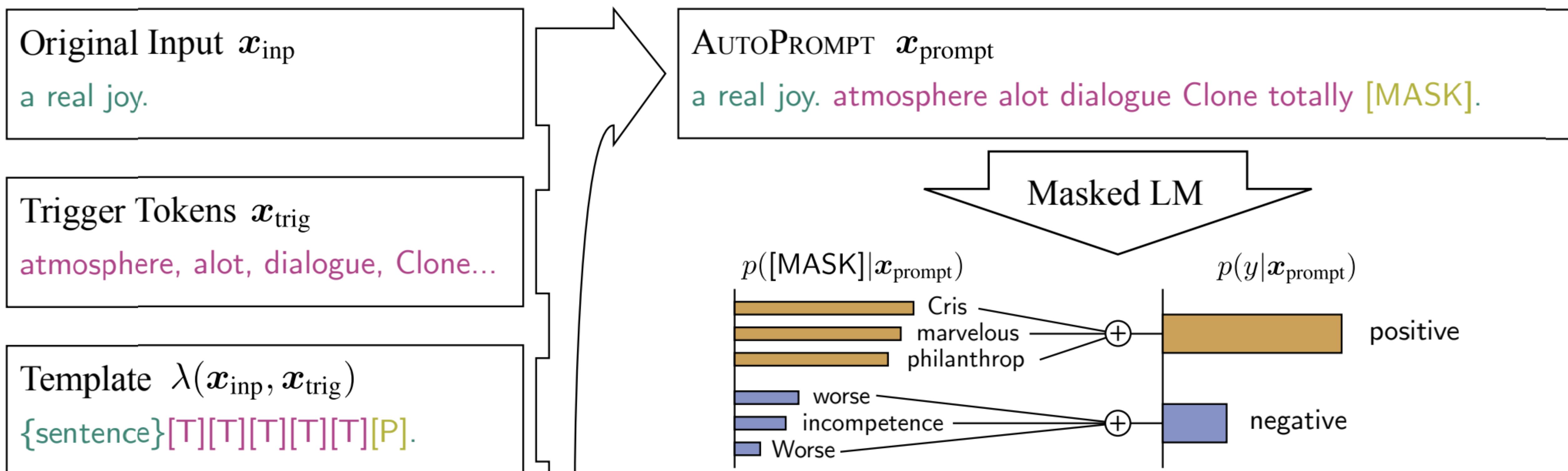


# Prompt Optimization

- ▶ A number of methods exist for searching over prompts
  - ▶ **Corpus-based methods:**
    - ▶ Find similar examples in the pre-training text (Jiang et al. 2019)
  - ▶ **Optimization-based:**
    - ▶ Use gradients or black-box optimization, e.g. RL (Wallace et al. 2019; Shin et al. 2020)
    - ▶ Most of these do not lead to dramatically better results than doing some manual engineering/hill-climbing (and they may be computationally intensive)
- ▶ Nevertheless, the choice of prompt *is* very important for zero-shot settings!

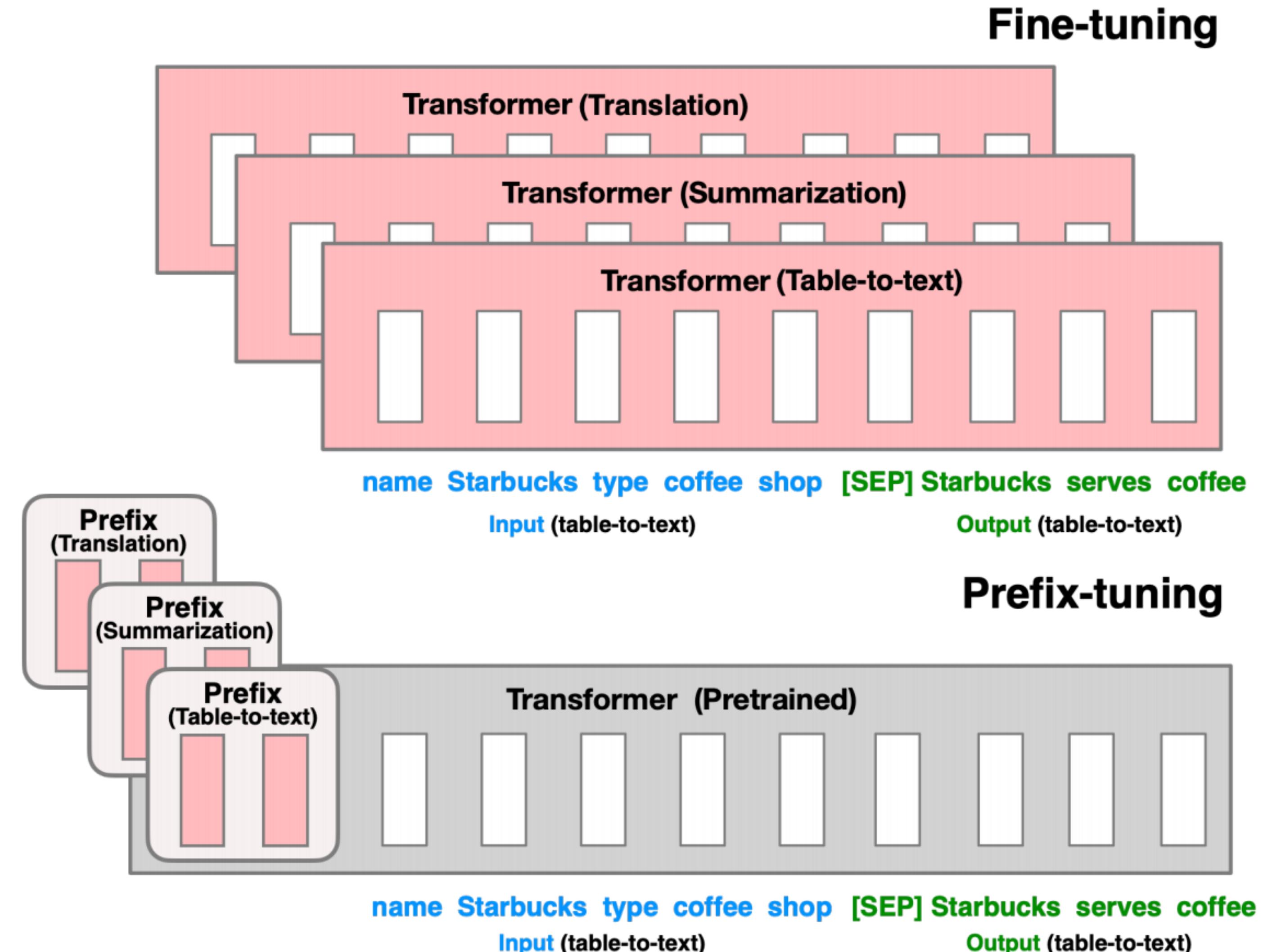
# Gradient-Based Search

- Automatically optimize arbitrary prompts based on existing words

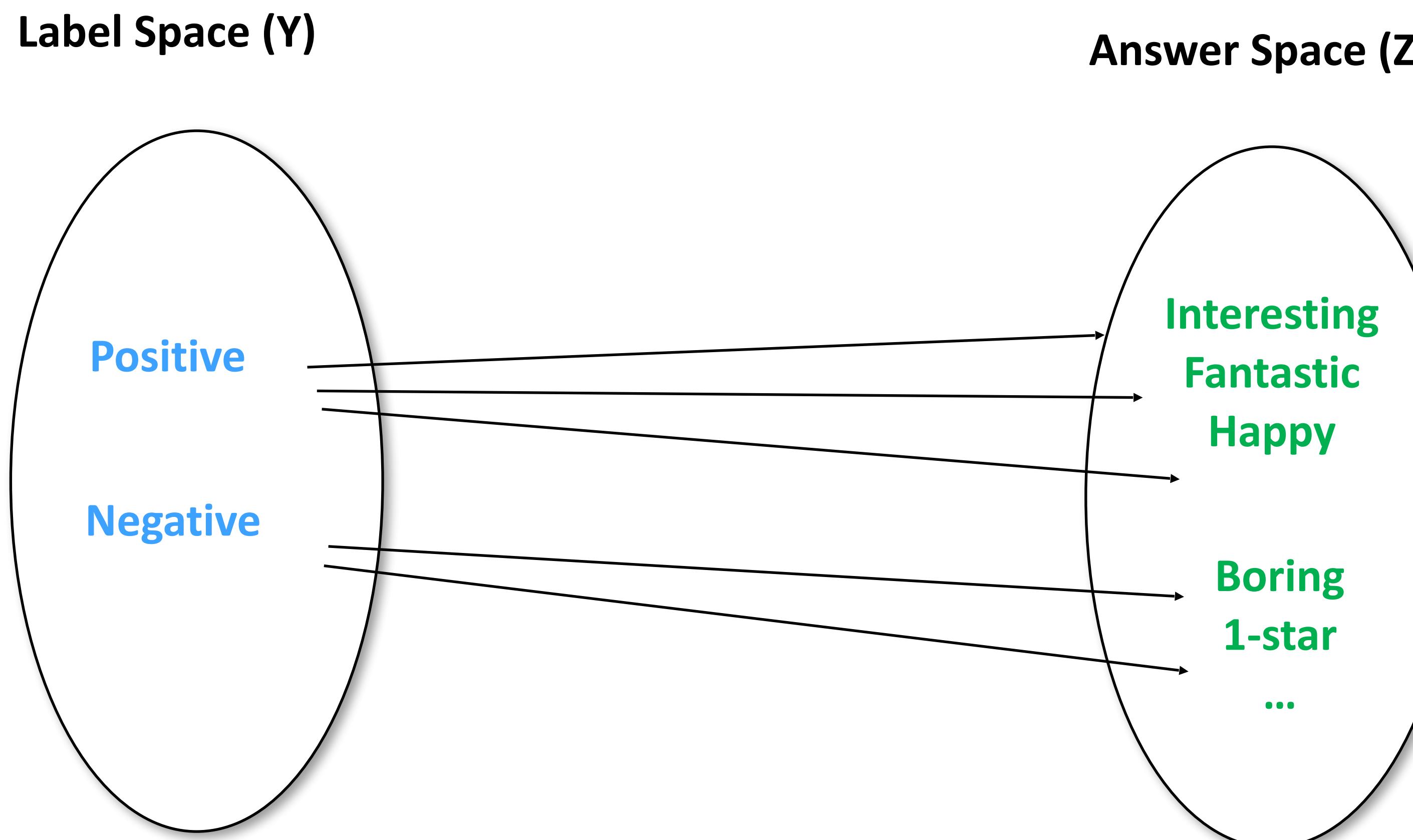


# Prefix/Prompt Tuning

- Optimize the embeddings of a prompt, instead of the words.
- "Prompt Tuning" optimizes only the embedding layer, "Prefix Tuning" optimizes prefix of all layers

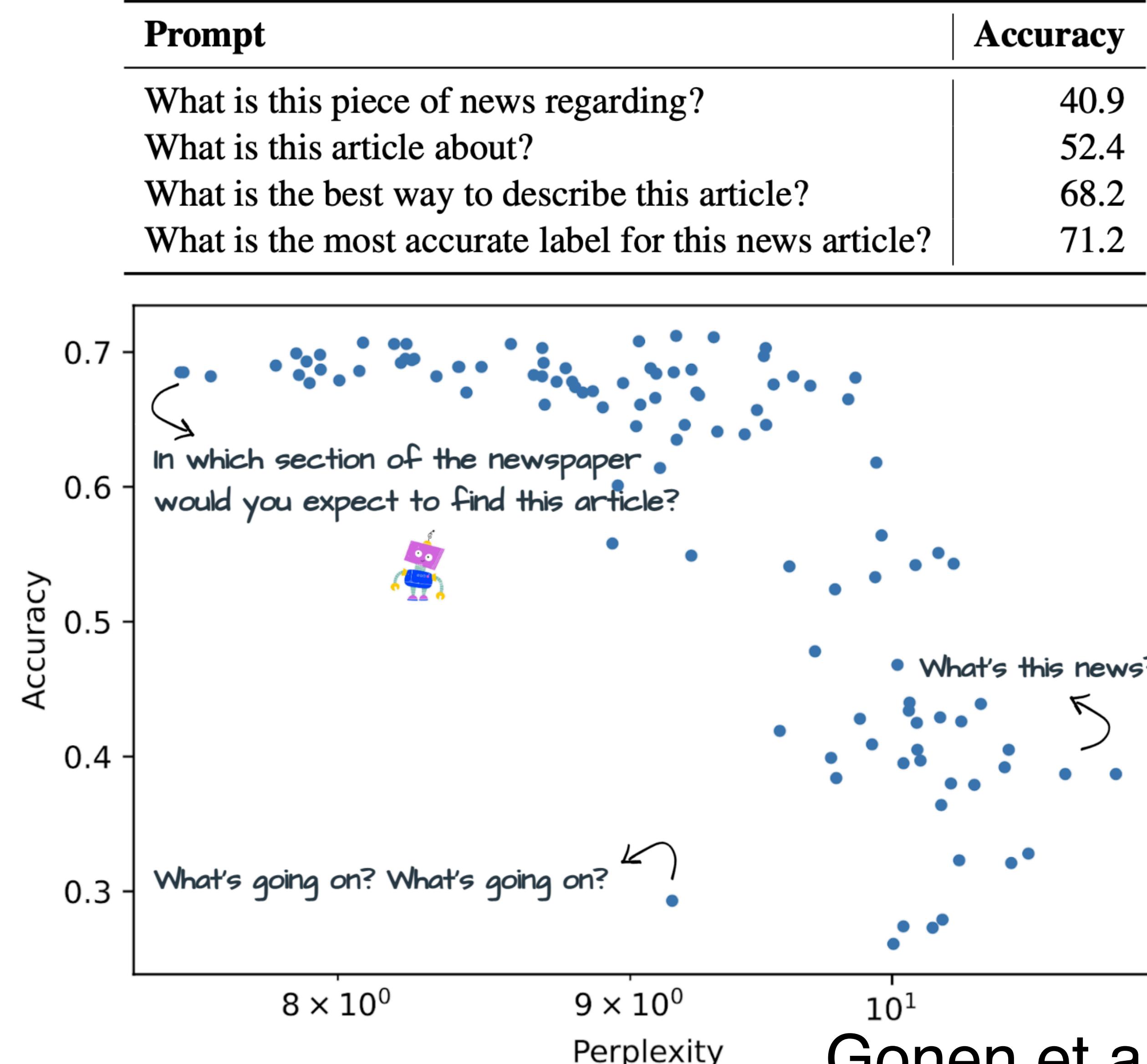


# Answer Mapping



# Variability in Prompts

- ▶ **Plot:** large number of prompts produced by {manual writing, paraphrasing, backtranslation}
- ▶ **x-axis:** perplexity of the prompt. How natural is it? How much does it appear in the pre-training data?
- ▶ **y-axis:** task performance



# Variability in Prompts

| Task            | Perplexity-score corr. |          | Perplexity-acc corr. |          | Avg Acc | Acc 50% |
|-----------------|------------------------|----------|----------------------|----------|---------|---------|
|                 | Pearson                | Spearman | Pearson              | Spearman |         |         |
| Antonyms        | **-0.41                | **-0.53  | -                    | -        | -       | -       |
| GLUE Cola       | -0.15                  | -0.14    | -0.04                | -0.02    | 47.7    | 57.1    |
| Newspop         | *-0.24                 | **-0.26  | *-0.20               | -0.18    | 66.4    | 72.9    |
| AG News         | **-0.63                | **-0.68  | **-0.77              | **-0.81  | 57.5    | 68.7    |
| IMDB            | **0.35                 | **0.40   | 0.14                 | *0.20    | 86.2    | 91.0    |
| DBpedia         | **-0.50                | **-0.44  | **-0.51              | **-0.42  | 46.7    | 55.2    |
| Emotion         | -0.14                  | -0.19    | **-0.30              | **-0.32  | 16.4    | 23.0    |
| Tweet Offensive | *-0.19                 | 0.07     | 0.18                 | *0.23    | 51.3    | 55.8    |

- OPT-175B: average of best 50% of prompts is much better than average over all prompts

Gonen et al. (2022)

# Few-shot Prompting

- ▶ Form “training examples” from  $(x, y)$  pairs, verbalize them (can be lighter-weight than zero-shot verbalizer)
- ▶ Input to GPT-3:  $v(x_1) v(y_1) v(x_2) v(y_2) \dots v(x_{\text{test}})$

*Review: The cinematography was stellar; great movie!*

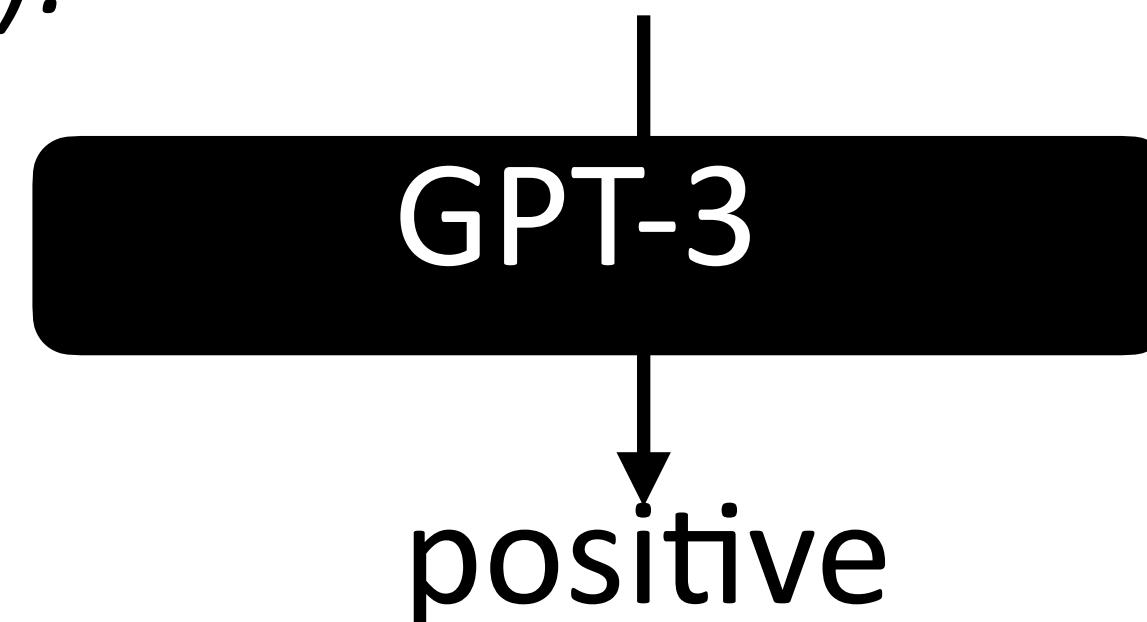
*Sentiment (positive or negative): positive*

*Review: The plot was boring and the visuals were subpar.*

*Sentiment (positive or negative): negative*

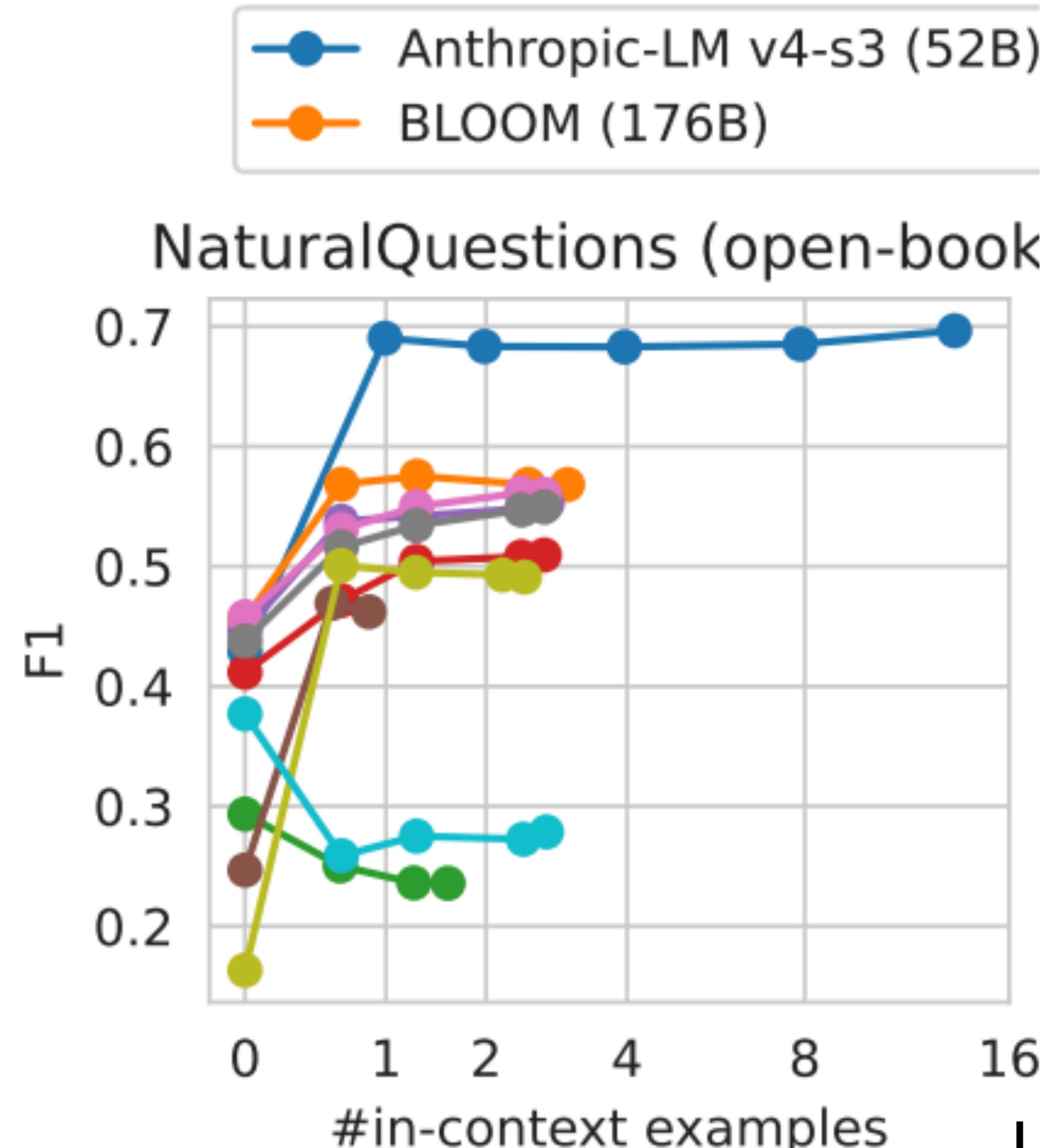
*Review: The movie's acting could've been better, but the visuals and directing were top-notch.*

*Sentiment (positive or negative):*



# Results: HELM

- ▶ So, how much better is few-shot compared to zero-shot?
- ▶ Each line is a different LM
- ▶ More in-context examples generally leads to better performance
- ▶ What do we see here?



# What can go wrong?

*Review: The movie was great!*

*Sentiment: positive*

*Review: I thought the movie was alright; I would've seen it again.*

*Sentiment: positive*

*Review: The movie was pretty cool!*

*Sentiment: positive*

*Review: Pretty decent movie!*

*Sentiment: positive*

*Review: The movie had good enough acting and the visuals were nice.*

*Sentiment: positive*

*Review: There wasn't anything the movie could've done better.*

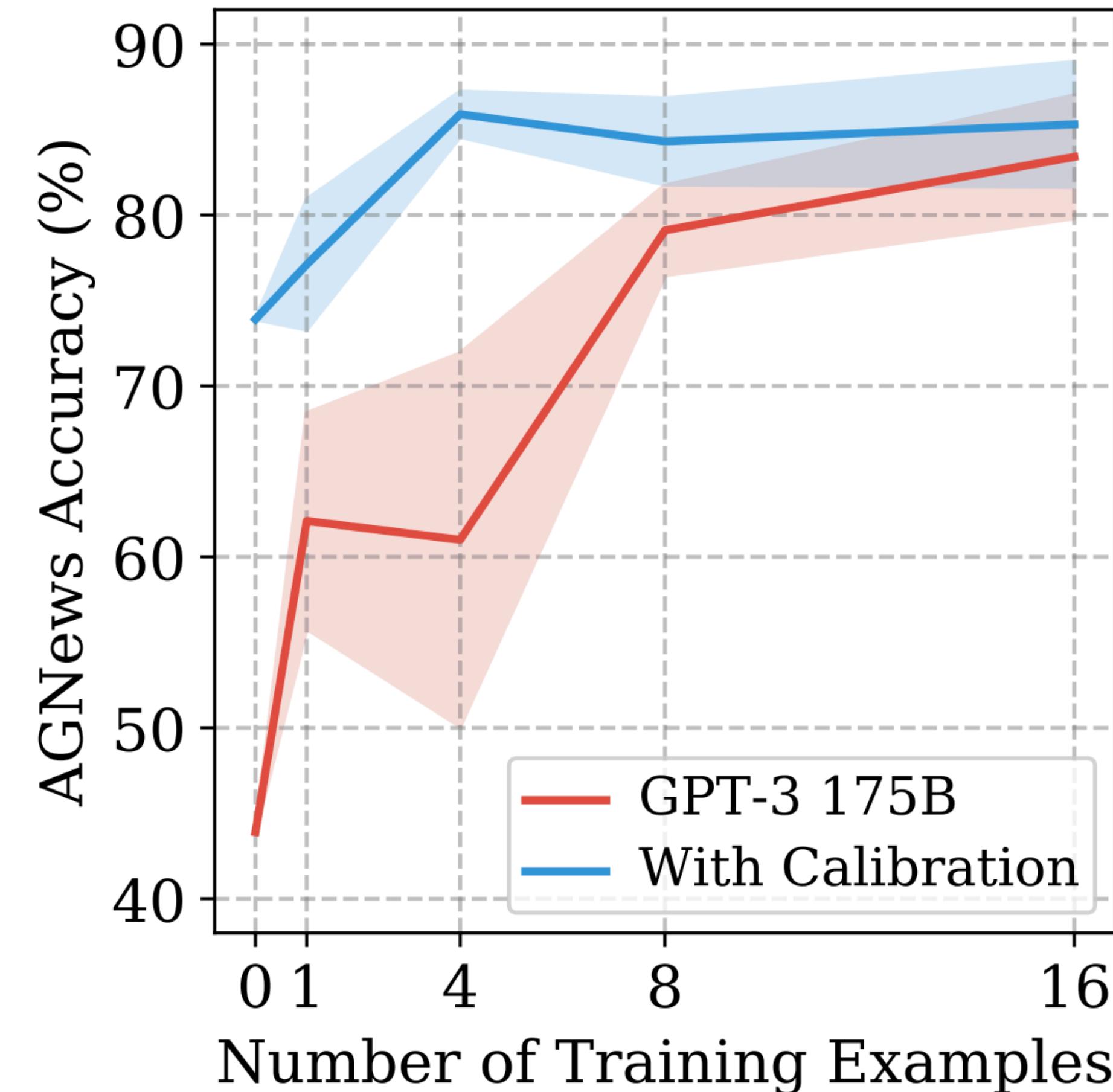
*Sentiment: positive*

*Review: Okay movie but could've been better.*

*Sentiment:* —  → *positive*

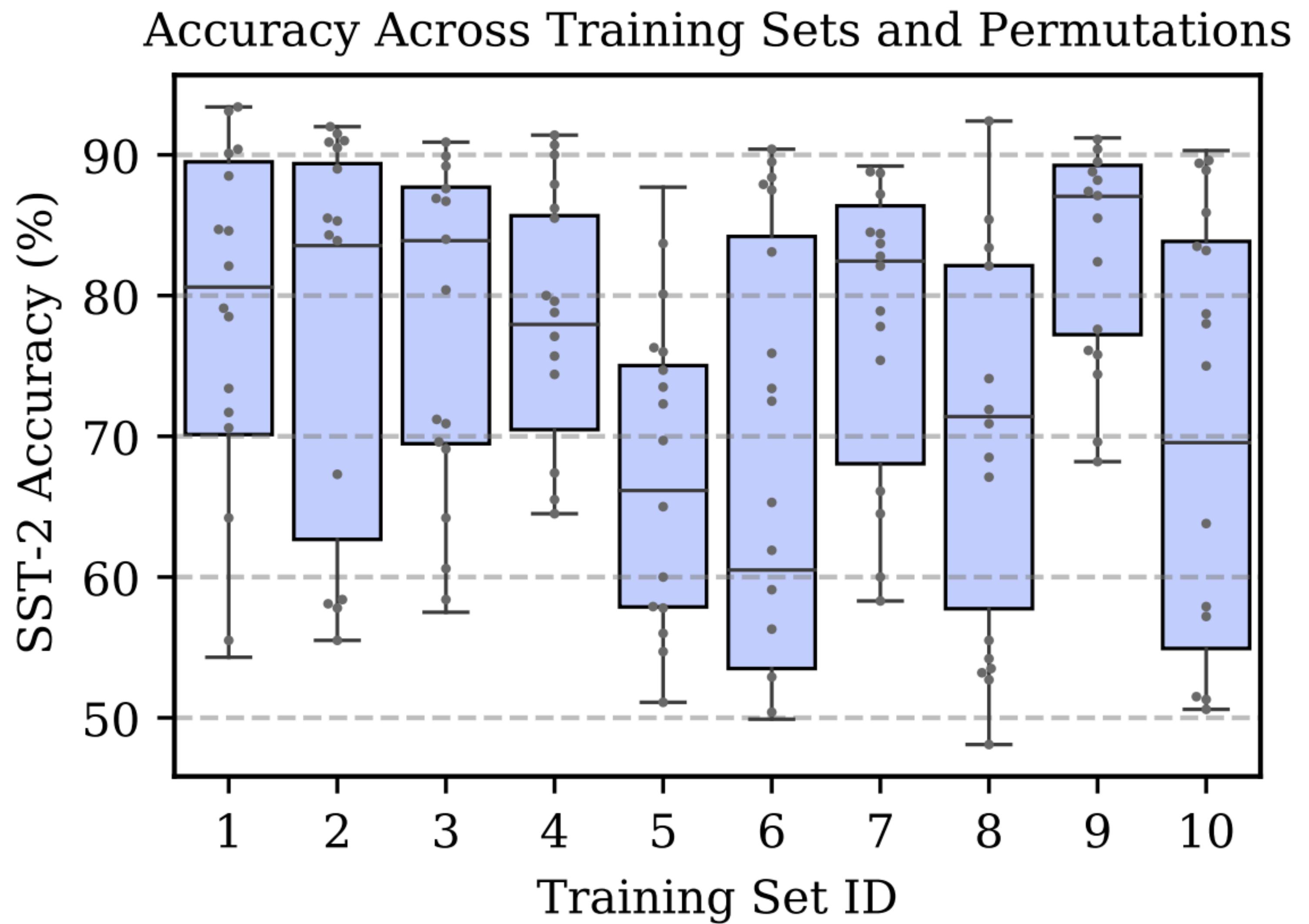
# What can go wrong?

- ▶ All one training label – model sees extremely skewed distribution
- ▶ What if we take random sets of training examples? There is quite a bit of variance on basic classification tasks
- ▶ Note: these results are with basic GPT-3 and not Instruct-tuned versions of the model. This issue has gotten a lot better



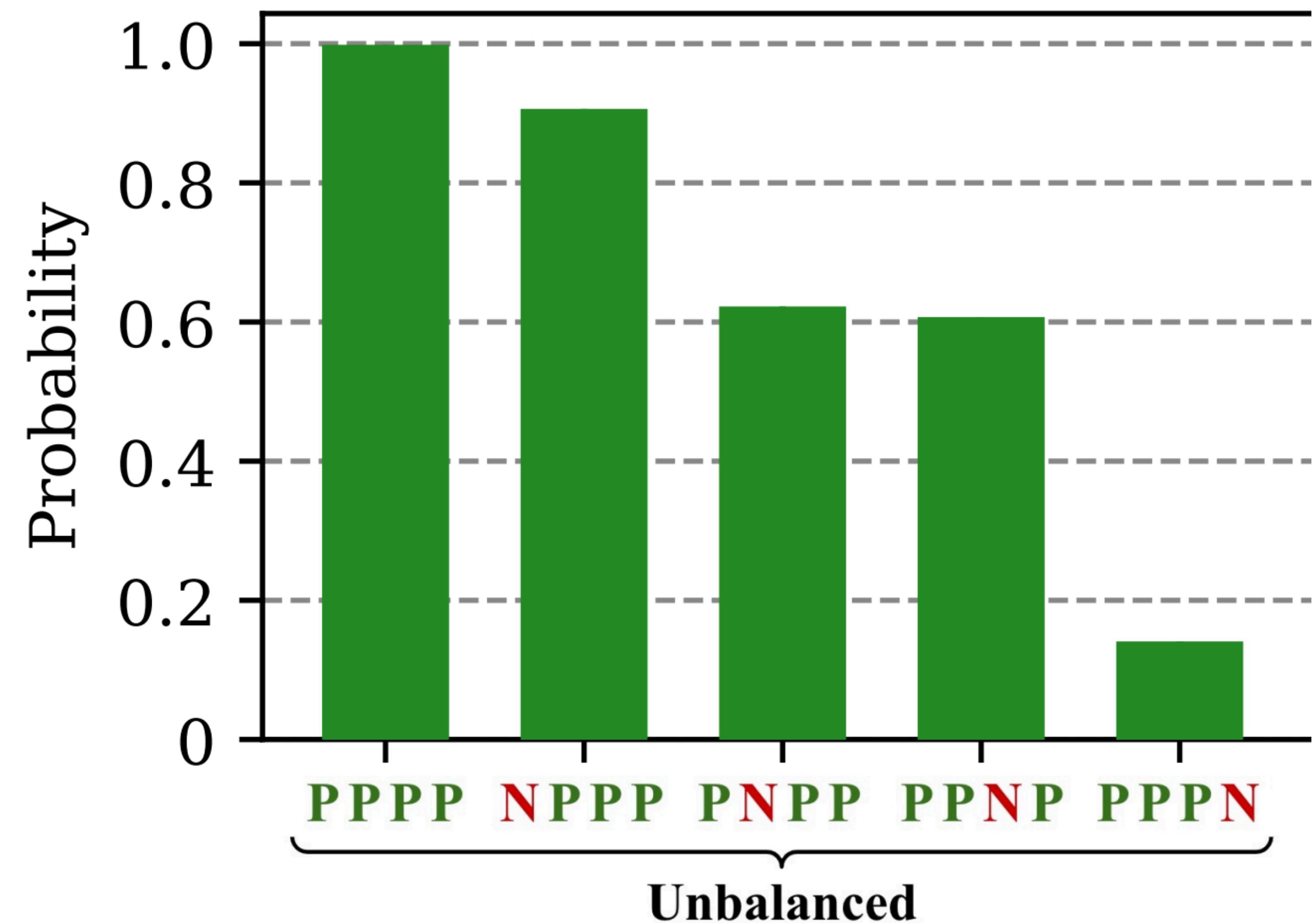
# What can go wrong?

- ▶ Varies even across permutations of training examples
- ▶ **x-axis:** different collections of train examples.  
**y-axis:** sentiment accuracy. Boxes represent results over different permutations of the data

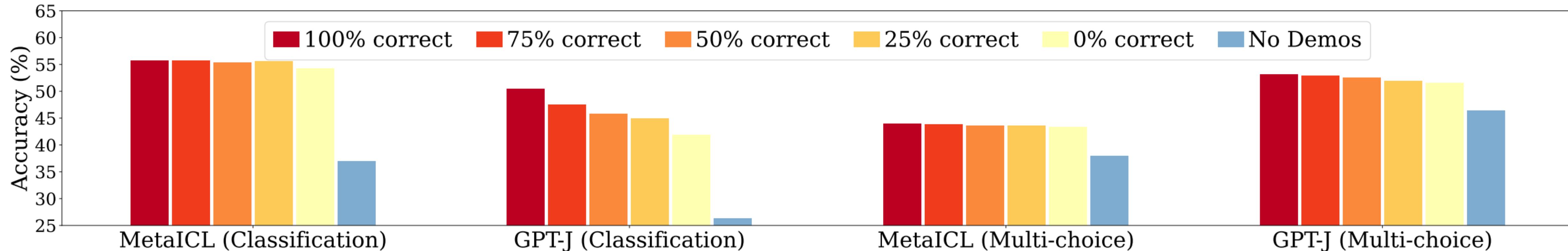


# What can go wrong?

- ▶ Having unbalanced training sets leads to high “default” probabilities of positive; that is, if we feed in a null  $\mathbf{x}_{\text{test}}$
- ▶ Solution: “calibrate” the model by normalizing by that probability of null  $\mathbf{x}_{\text{test}}$
- ▶ Leads to higher performance; not necessarily crucial with prompt-tuned models



# Rethinking Demonstrations



- ▶ Surprising result: Using random labels for few-shot examples does not substantially decrease performance??
- ▶ Other reasons few-shot learning might work:
  - ▶ Model has seen similar tasks/domains in pre-training; few-shot examples help it “retrieve” the task.
  - ▶ Few-shot examples help model learn I/O format and output labels.

# Chain-of-thought

- ▶ Typically a few-shot prompting technique where the in-context examples now contain explanations
- ▶ Answer is not generated in one go, but comes after an explanation that “talks through” the 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).

## StrategyQA

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

A: The density of a pear is about 0.6 g/cm<sup>3</sup>, 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.

# Step-by-Step

## (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.* ✓

- ▶ Prompt for zero-shot step-by-step reasoning: produces chains of thought without including demonstrations
- ▶ Separate prompt to extract the answer (“Therefore, the answer is \_\_\_\_”)

# Step-by-Step

| No. | Category    | Template                                                  | Accuracy    |
|-----|-------------|-----------------------------------------------------------|-------------|
| 1   | instructive | Let's think step by step.                                 | <b>78.7</b> |
| 2   |             | First, (*1)                                               | 77.3        |
| 3   |             | Let's think about this logically.                         | 74.5        |
| 4   |             | Let's solve this problem by splitting it into steps. (*2) | 72.2        |
| 5   |             | Let's be realistic and think step by step.                | 70.8        |
| 6   |             | Let's think like a detective step by step.                | 70.3        |
| 7   |             | Let's think                                               | 57.5        |
| 8   |             | Before we dive into the answer,                           | 55.7        |
| 9   |             | The answer is after the proof.                            | 45.7        |
| 10  | misleading  | Don't think. Just feel.                                   | 18.8        |
| 11  |             | Let's think step by step but reach an incorrect answer.   | 18.7        |
| 12  |             | Let's count the number of "a" in the question.            | 16.7        |
| 13  |             | By using the fact that the earth is round,                | 9.3         |
| 14  | irrelevant  | By the way, I found a good restaurant nearby.             | 17.5        |
| 15  |             | Abrakadabra!                                              | 15.5        |
| 16  |             | It's a beautiful day.                                     | 13.1        |
| -   |             | (Zero-shot)                                               | 17.7        |

# Self-Consistency

## Self-consistency

**Q:** If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

**A:** There are 3 cars in the parking lot already. 2 more arrive. Now there are  $3 + 2 = 5$  cars. The answer is 5.

...  
**Q:** Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder for \$2 per egg. How much does she make every day?

**A:**

Language model

Sample a diverse set of reasoning paths

She has  $16 - 3 - 4 = 9$  eggs left. So she makes  $\$2 * 9 = \$18$  per day.

This means she sells the remainder for  $\$2 * (16 - 4 - 3) = \$26$  per day.

She eats 3 for breakfast, so she has  $16 - 3 = 13$  left. Then she bakes muffins, so she has  $13 - 4 = 9$  eggs left. So she has  $9 \text{ eggs} * \$2 = \$18$ .

Marginalize out reasoning paths to aggregate final answers

The answer is \$18.

The answer is \$26.

The answer is \$18.

- ▶ Ensembling across multiple outputs (either zero-shot or few-shot)
- ▶ GSM8k: 56.5 -> 74.4, 5% gains on several other math datasets, lower gains on text tasks

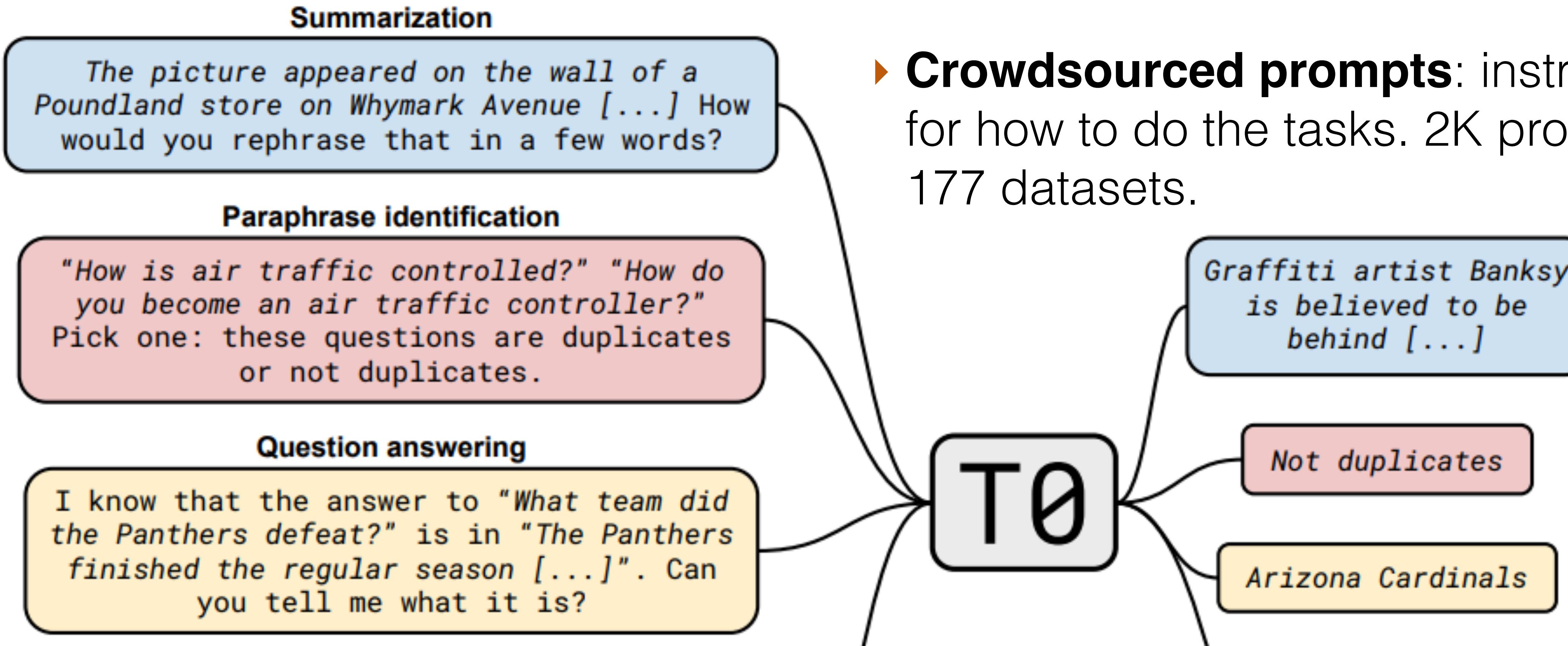
# **Multi-Task Model Engineering: Instruction Tuning**

# Instruction Tuning

- ▶ We want to optimize models for  $P(\text{answer} \mid \text{prompt, input})$ , but they're learned on a basic language modeling objective
- ▶ One solution: treat the basic language modeling as pre-training, then fine-tune them on what we care about
- ▶ Two versions of this:
  - ▶ **Instruction tuning:** supervised fine-tuning on data derived from many NLP tasks
  - ▶ **Reinforcement learning from human feedback (RLHF):** RL to improve human judgments of how good the outputs are

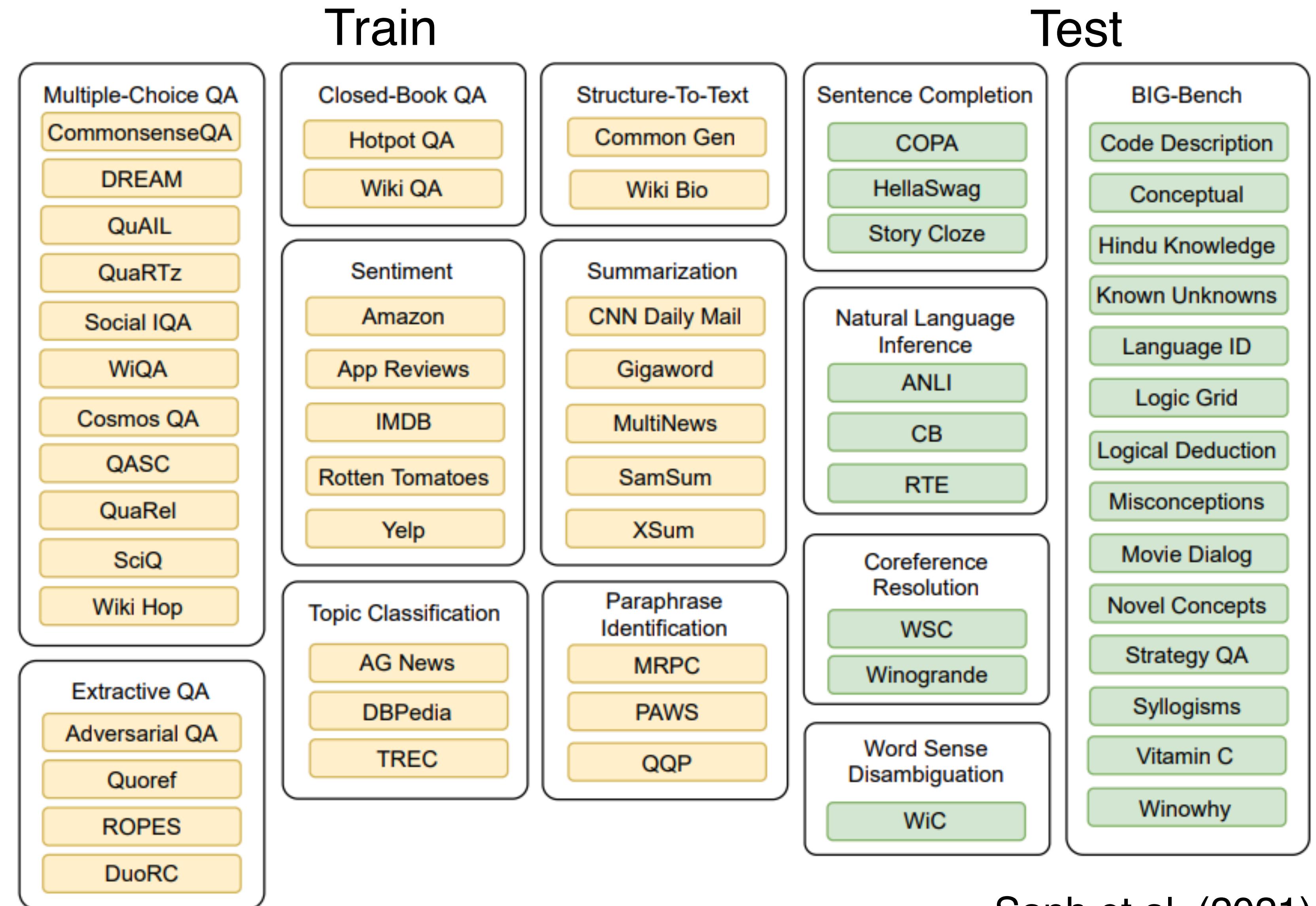
# Task Generalization: T0

- ▶ T0: tries to deliver on the goal of T5 and do many tasks with one model
- ▶ **Crowdsourced prompts:** instructions for how to do the tasks. 2K prompts for 177 datasets.

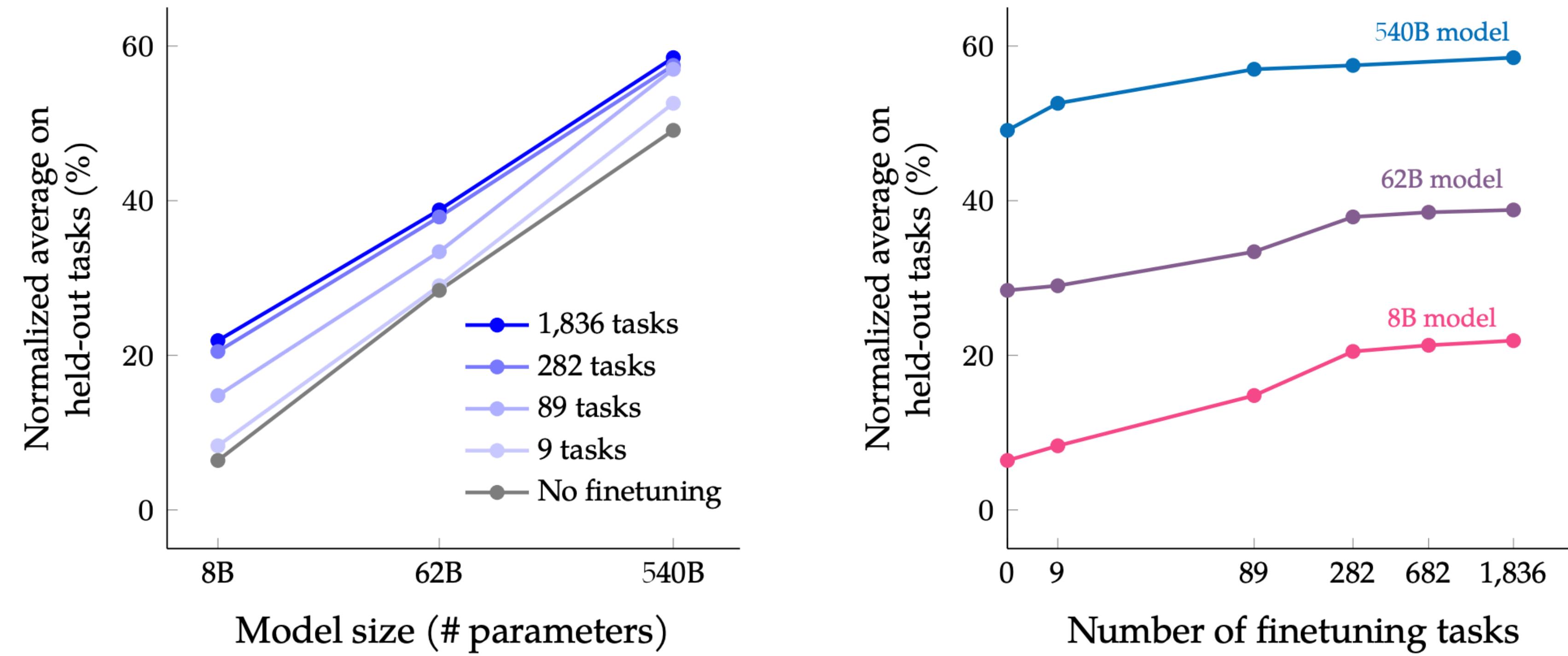


# Task Generalization

- ▶ Pre-train: T5 task
- ▶ Train: a collection of tasks with prompts. **This uses existing labeled training data**
- ▶ Test: a new task specified only by a new prompt. **No training data in this task**

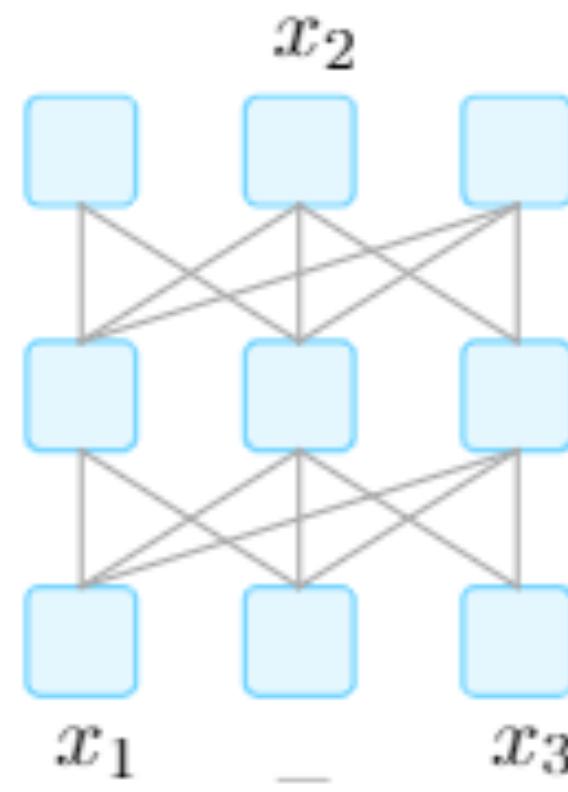


# Flan-T5

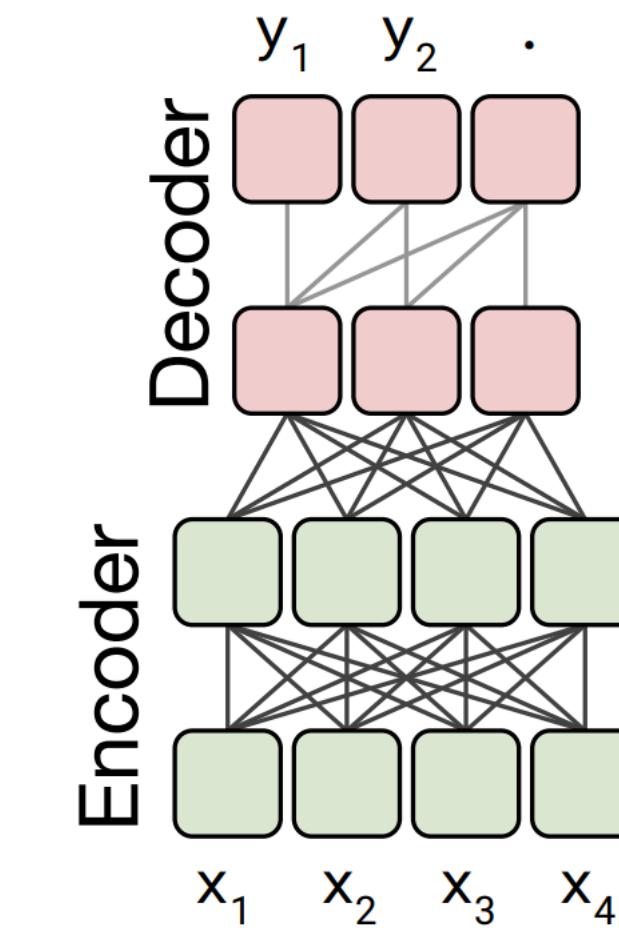


- What happens if you're Google and can scale the model and the # of tasks?
- Flan-T5-11B: Best model at the ~10B scale for few-shot prompting, also very good choice for fine-tuning. You can use this in your projects.

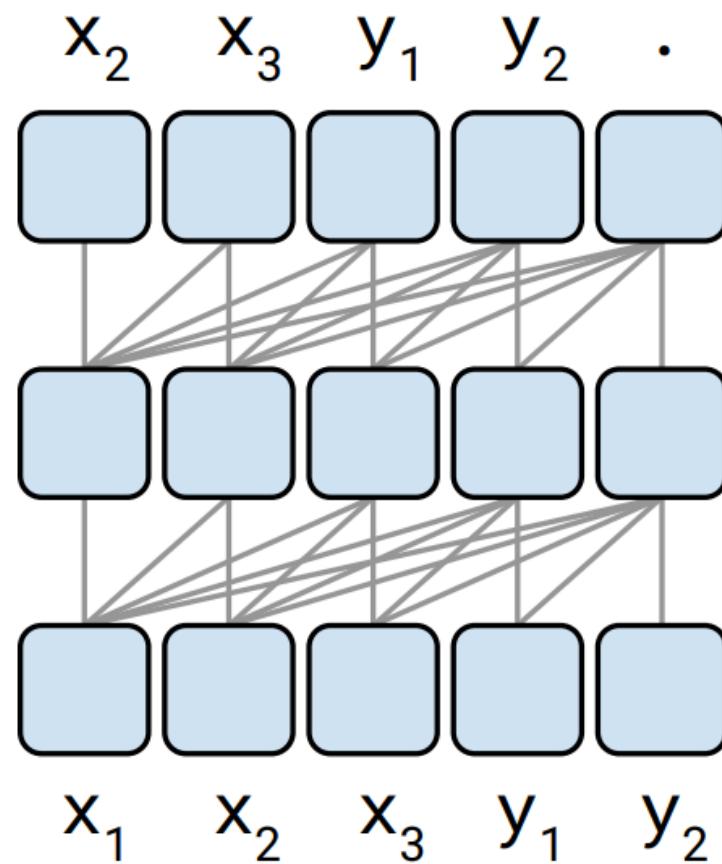
# Model Zoo



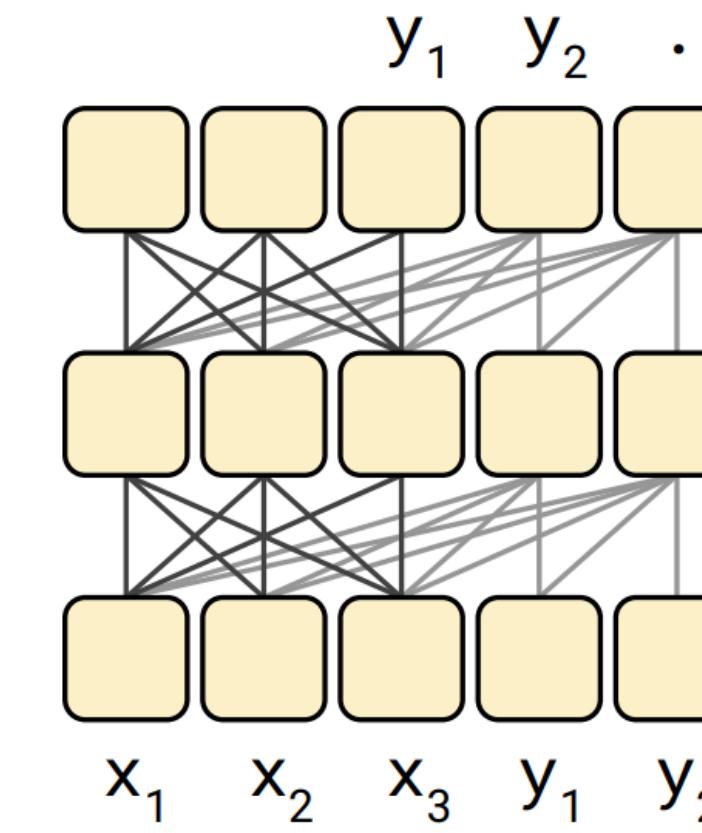
Masked Language Model  
(e.g. BERT)



Encoder-Decoder  
(e.g. BART, T5)



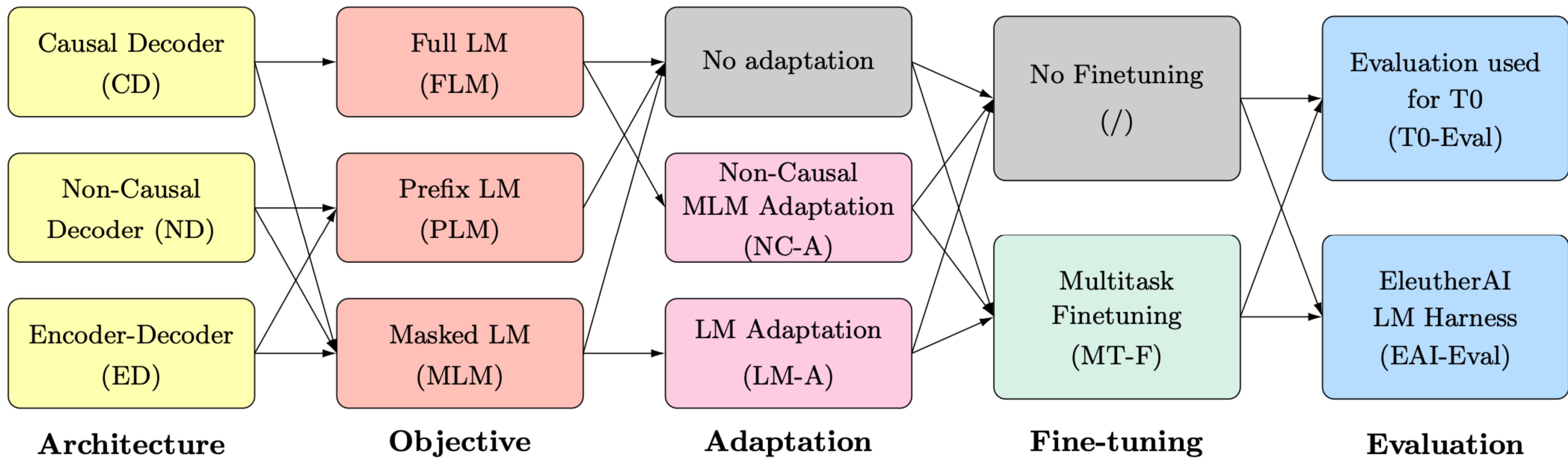
(Causal) Language Model  
(e.g. GPT)



Prefix Language Model  
(e.g. UniLM, UL2)

Figures from  
T5; Raffel et al. 2019

# Training Zoo



- ▶ Recent work exploring the space of architectures and training objectives:  
BigScience (Wang et al. 2022); UL2 (Tay et al. 2022)

# Questions?