CS11-711 Advanced NLP

Prompting and Instruction Tuning

Daniel Fried

Carnegie Mellon University
Language Technologies Institute

Site

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 kernelized support vector machine (SVM)
  - 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**:
  - CNN 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

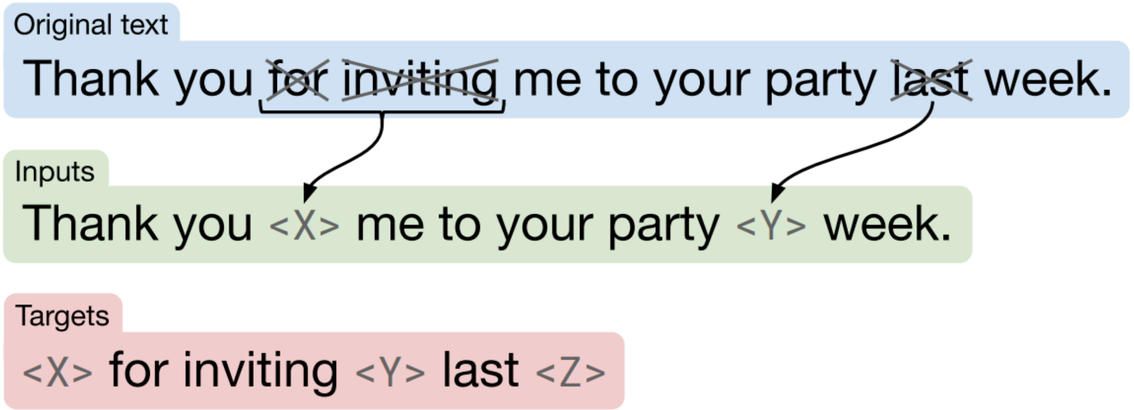
- **Paradigm**: Pre-train, Instruction-tuning/RL from human feedback, prompt.
- **Date**: 2021-Now
- **Characteristic**:
  - Train an LLM to be able to do multiple tasks, conditioned on a task description
  - Doesn’t require fine-tuning
  - Makes the model more robust to prompts
  - Requires rich data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

Objective Engineering:
T5 Recap

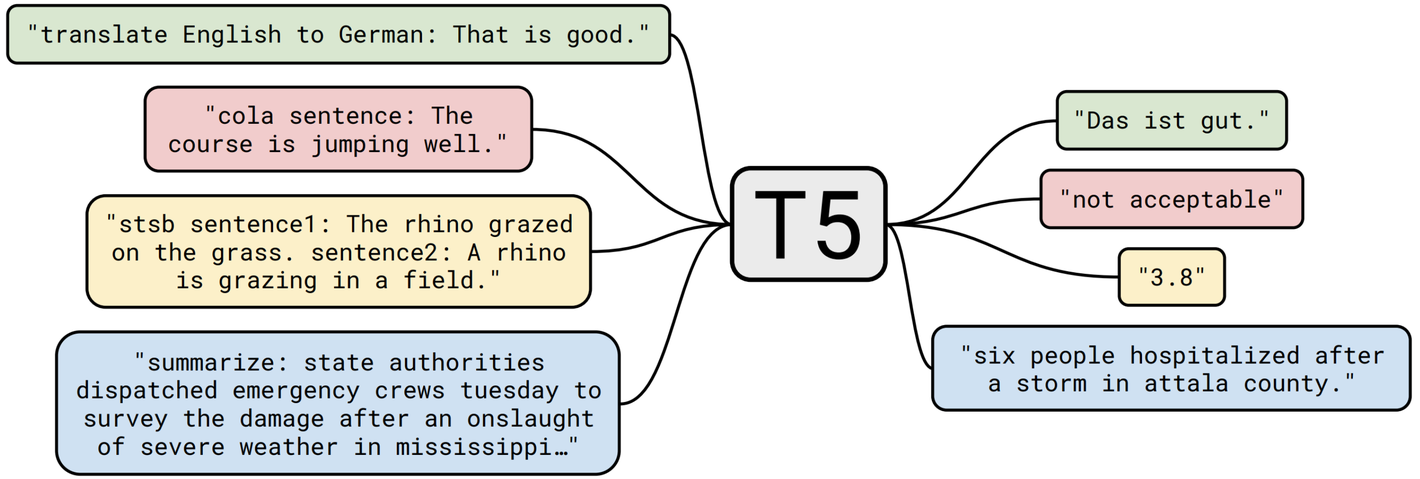
T5

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

Pre-training:



Fine-tuning:



Raffel et al. (2019)

UnifiedQA

Multiple choice

Yes/no

MC	**Dataset**	MCTest
	Input	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. ...
	Output	The big kid
YN	**Dataset**	BoolQ
	Input	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 ...
	Output	no

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

Khashabi et al. (2020)

UnifiedQA

AB	**Dataset**	NarrativeQA
	Input	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.'' ...
	Output	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

Khashabi et al. (2020)

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)

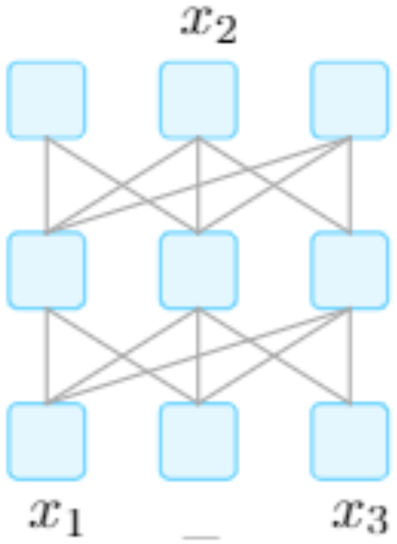
- If we have a strong enough pre-trained model and train on enough tasks, can we generalize to new tasks?

- How do we specify those new tasks if they're not close to tasks we've already run on?

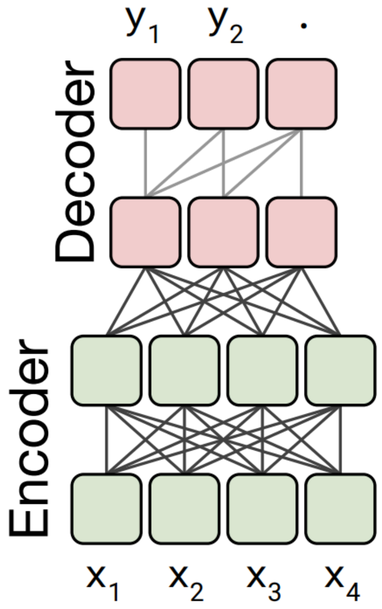
- Answer: prompting. But to do that well, we'll need to scale up further

Prompt Engineering:
LLMs and GPT

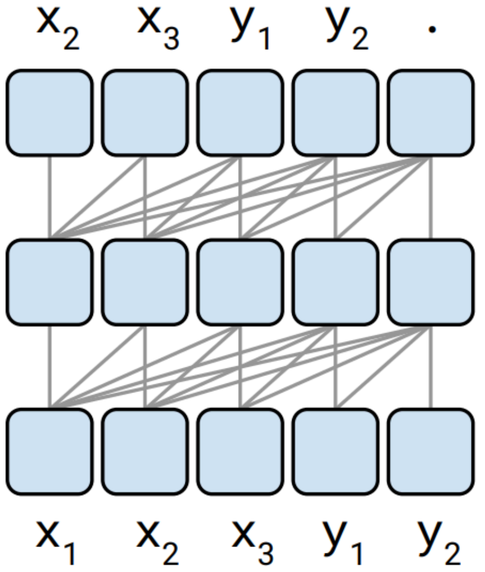
Model Zoo



Masked Language Model
(e.g. BERT)



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



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

Figures from
T5; Raffel et al. 2019

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

approximate size of BERT

GPT-2

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
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

Radford et al. (2019)

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)

https://syncedreview.com/2019/06/27/the-staggering-cost-of-training-sota-ai-models/

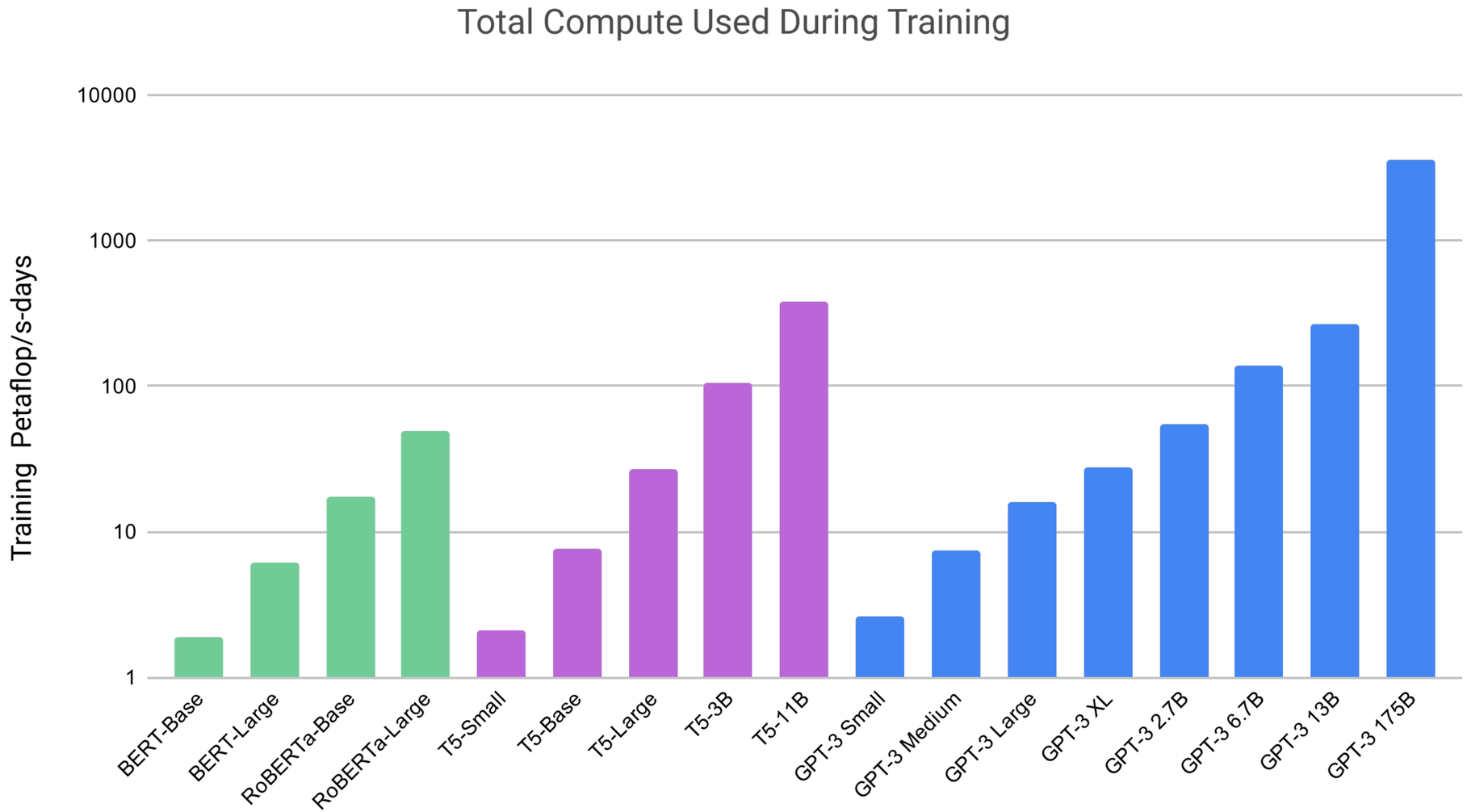
Prompts

- 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?

Brown et al. (2020)

Pushing the Limits: GPT-3

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



Brown et al. (2020)

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

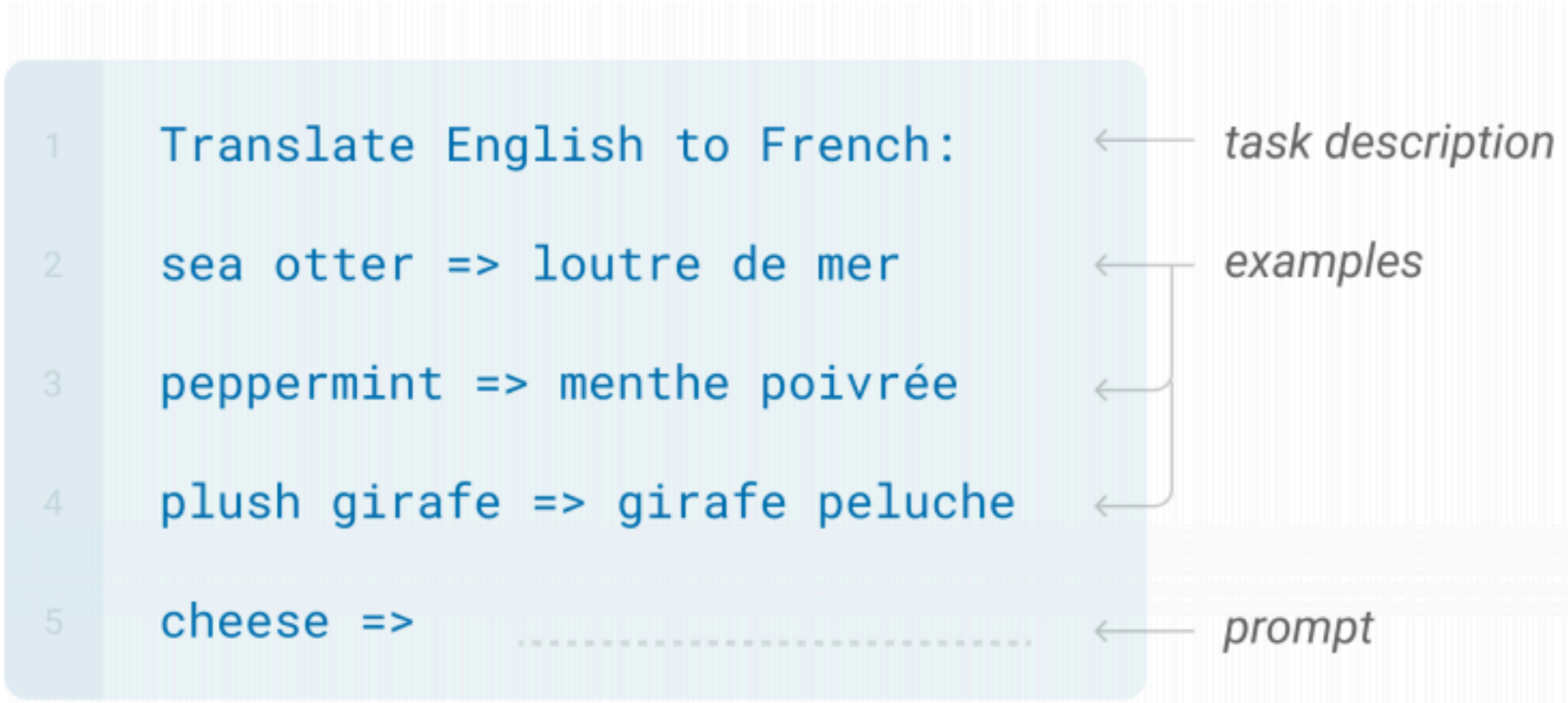
- **This is super expensive with 175B parameters**

Brown et al. (2020)

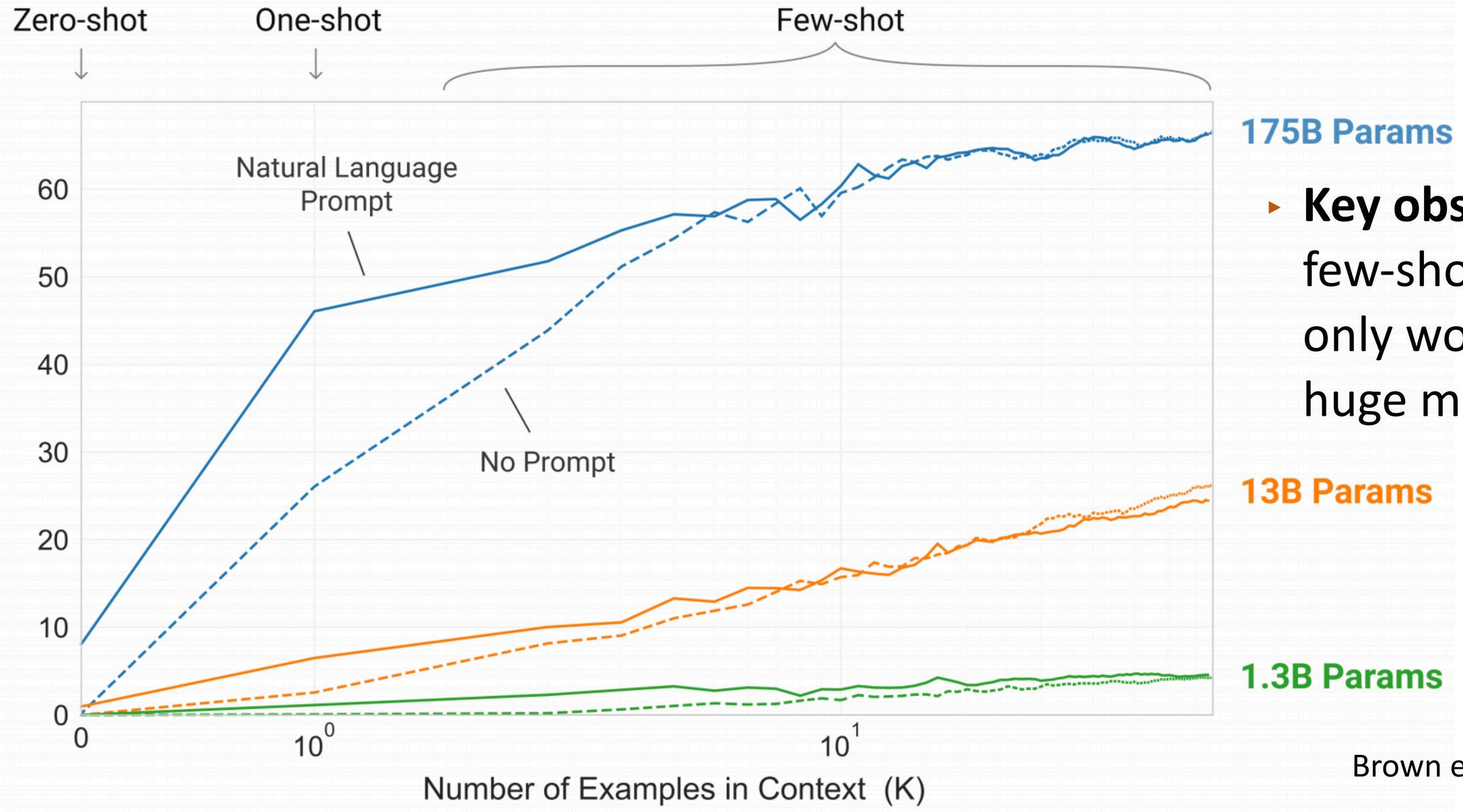
GPT-3: Few-shot Learning

- 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*”)



GPT-3



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

Brown et al. (2020)

GPT-3

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
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 Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
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!

Brown et al. (2020)

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

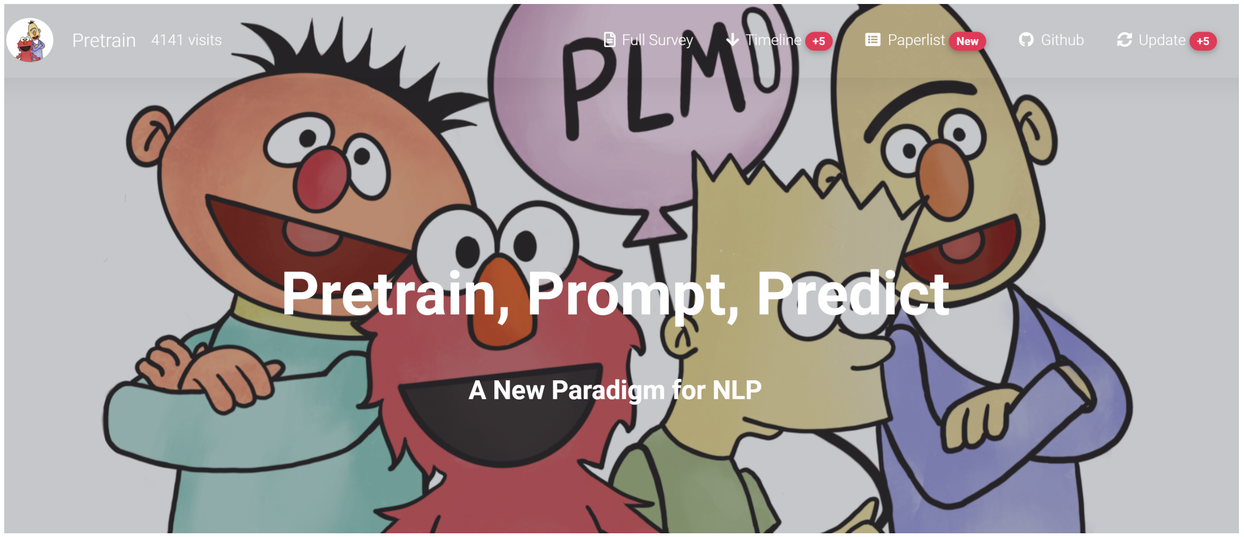
Weizhe Yuan
Carnegie Mellon University
weizhey@cs.cmu.edu

Jinlan Fu
National University of Singapore
jinlanjonna@gmail.com

Zhengbao Jiang
Carnegie Mellon University
zhengbaj@cs.cmu.edu

Hiroaki Hayashi
Carnegie Mellon University
hiroakih@cs.cmu.edu

Graham Neubig
Carnegie Mellon University
gneubig@cs.cmu.edu



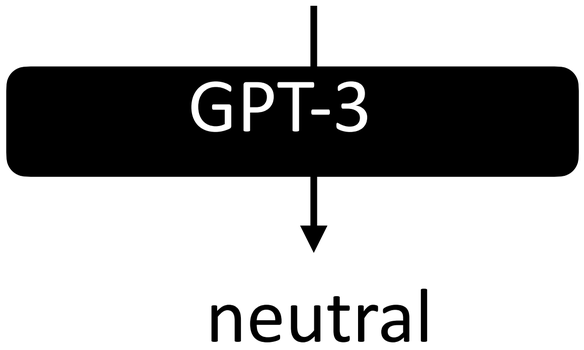
Zero-shot Prompting

- Single unlabeled datapoint ***x***, want to predict label *y*

  x = *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



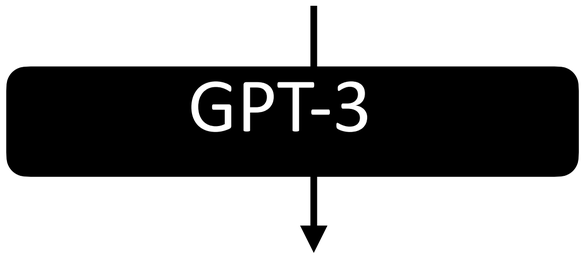
Zero-shot Prompting

- Single unlabeled datapoint ***x***, want to predict label *y*

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

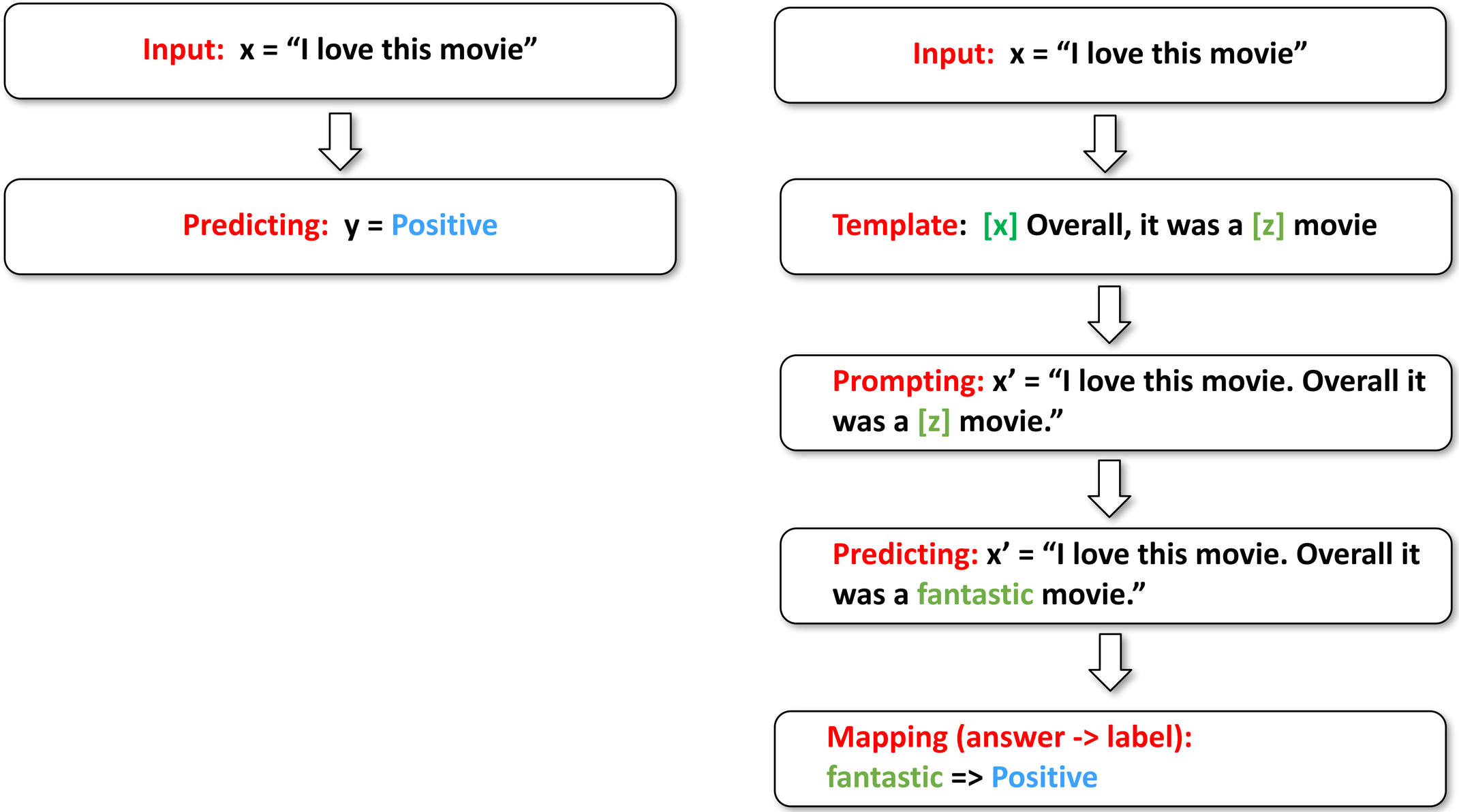
- Wrap ***x*** in a template we call a verbalizer ***v***

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 (either using gradients or black-box optimization)
- 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

Original Input $\boldsymbol{x}_{\text{inp}}$

a real joy.

Trigger Tokens $\boldsymbol{x}_{\text{trig}}$

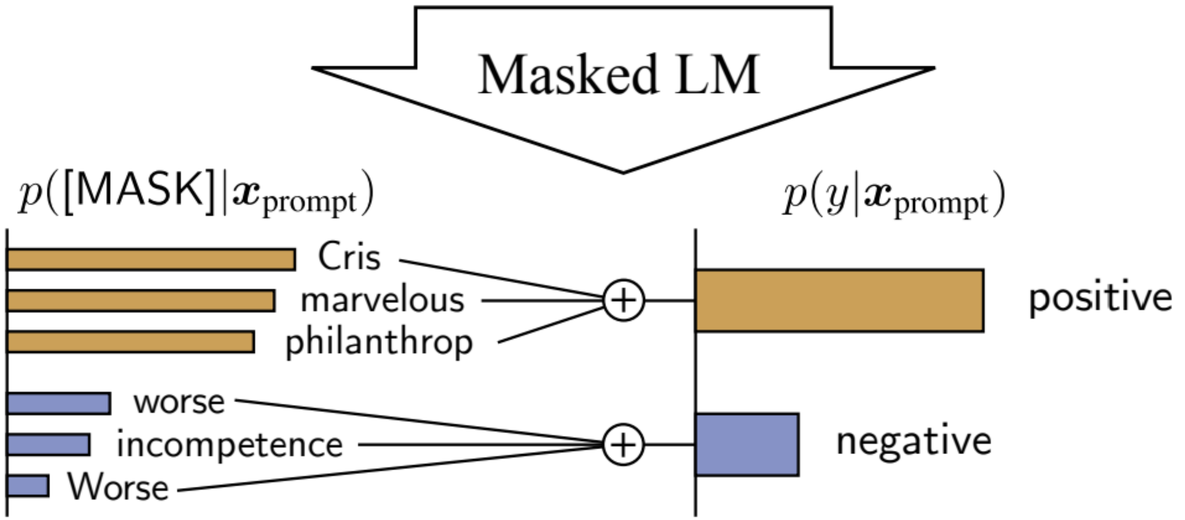
atmosphere, alot, dialogue, Clone...

Template $\lambda(\boldsymbol{x}_{\text{inp}}, \boldsymbol{x}_{\text{trig}})$

{sentence}[T][T][T][T][T][P].

AUTOPROMPT $\boldsymbol{x}_{\text{prompt}}$

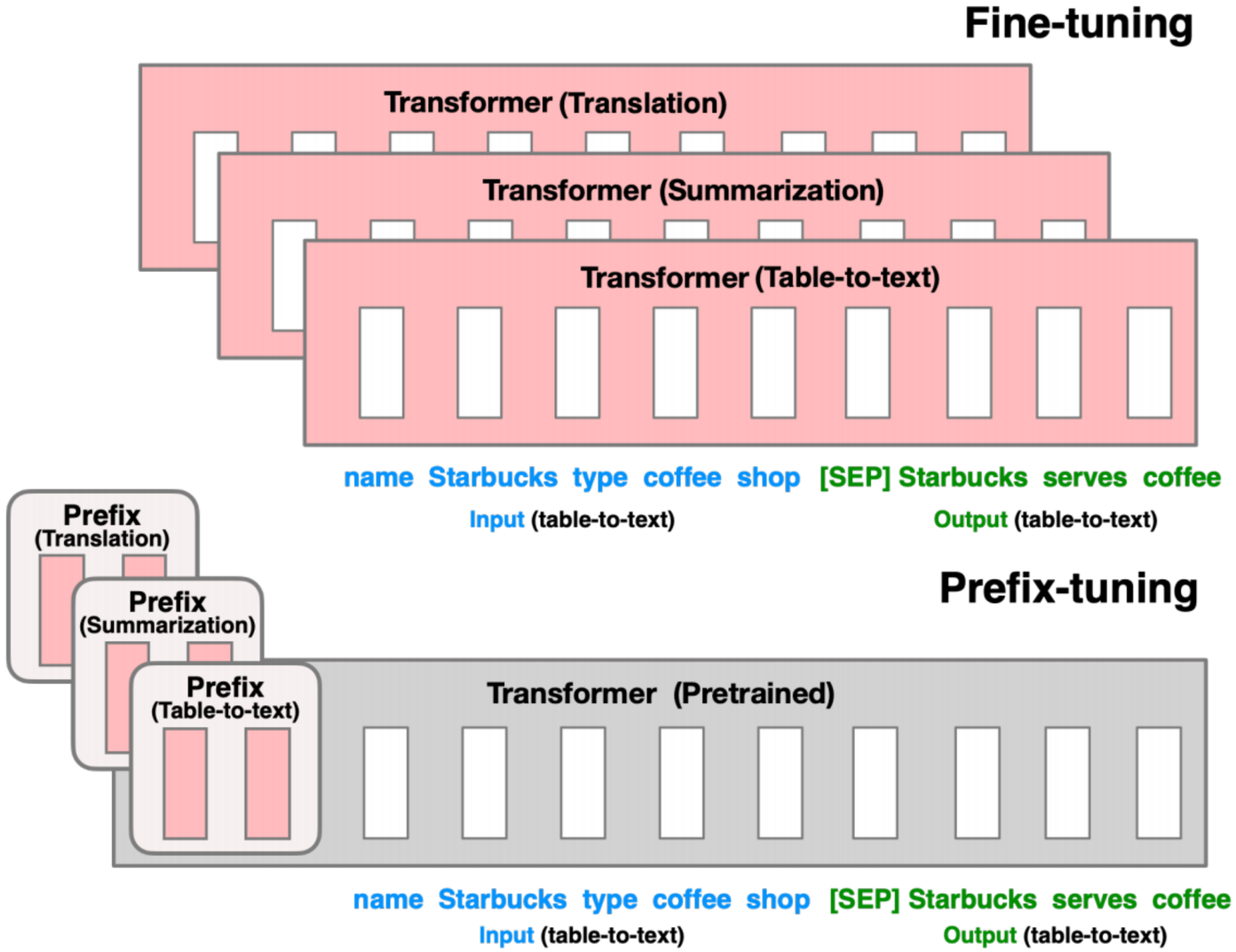
a real joy. atmosphere alot dialogue Clone totally [MASK].



Shin et al. (2020)

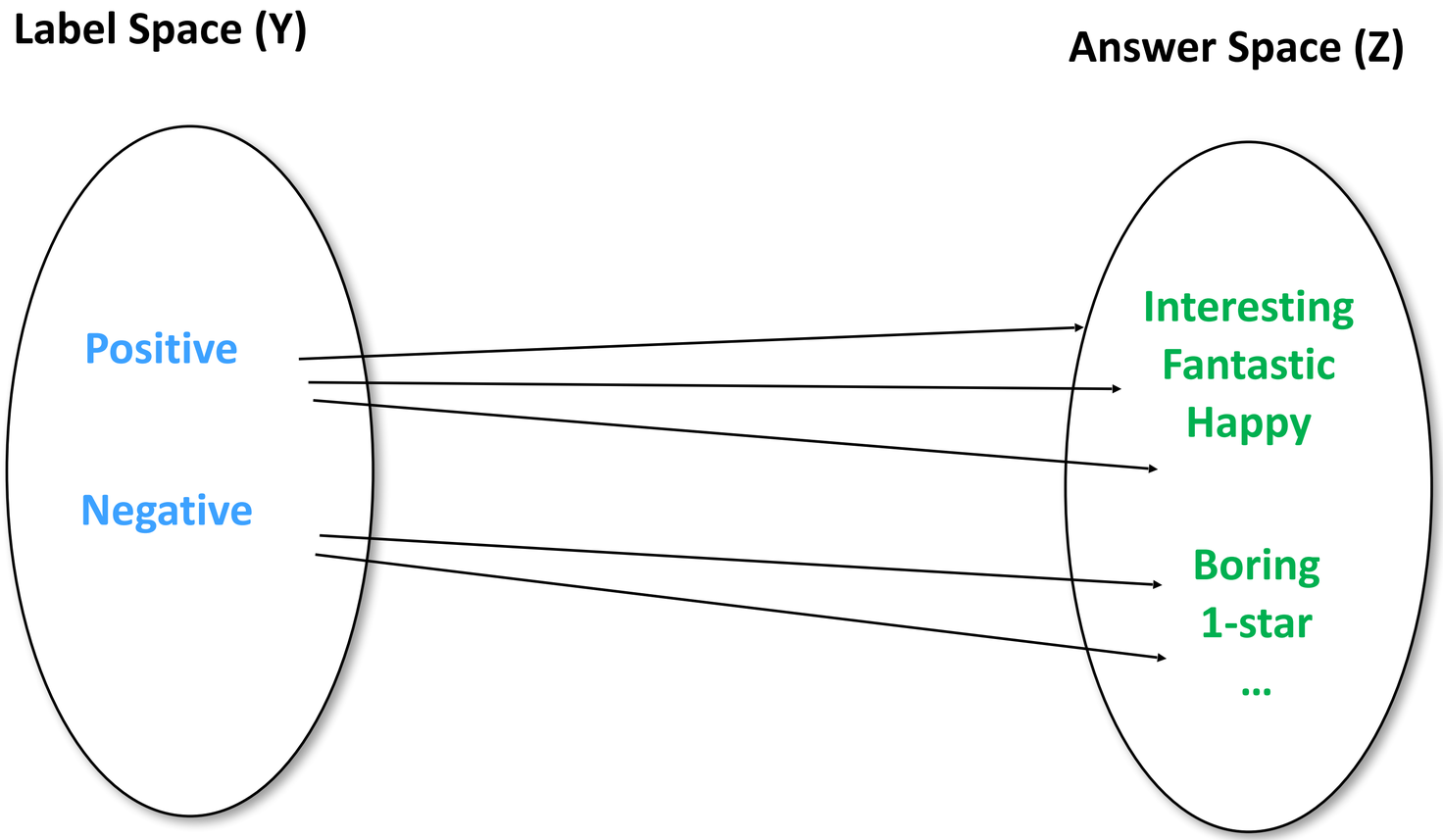
Prefix/Prompt Tuning

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



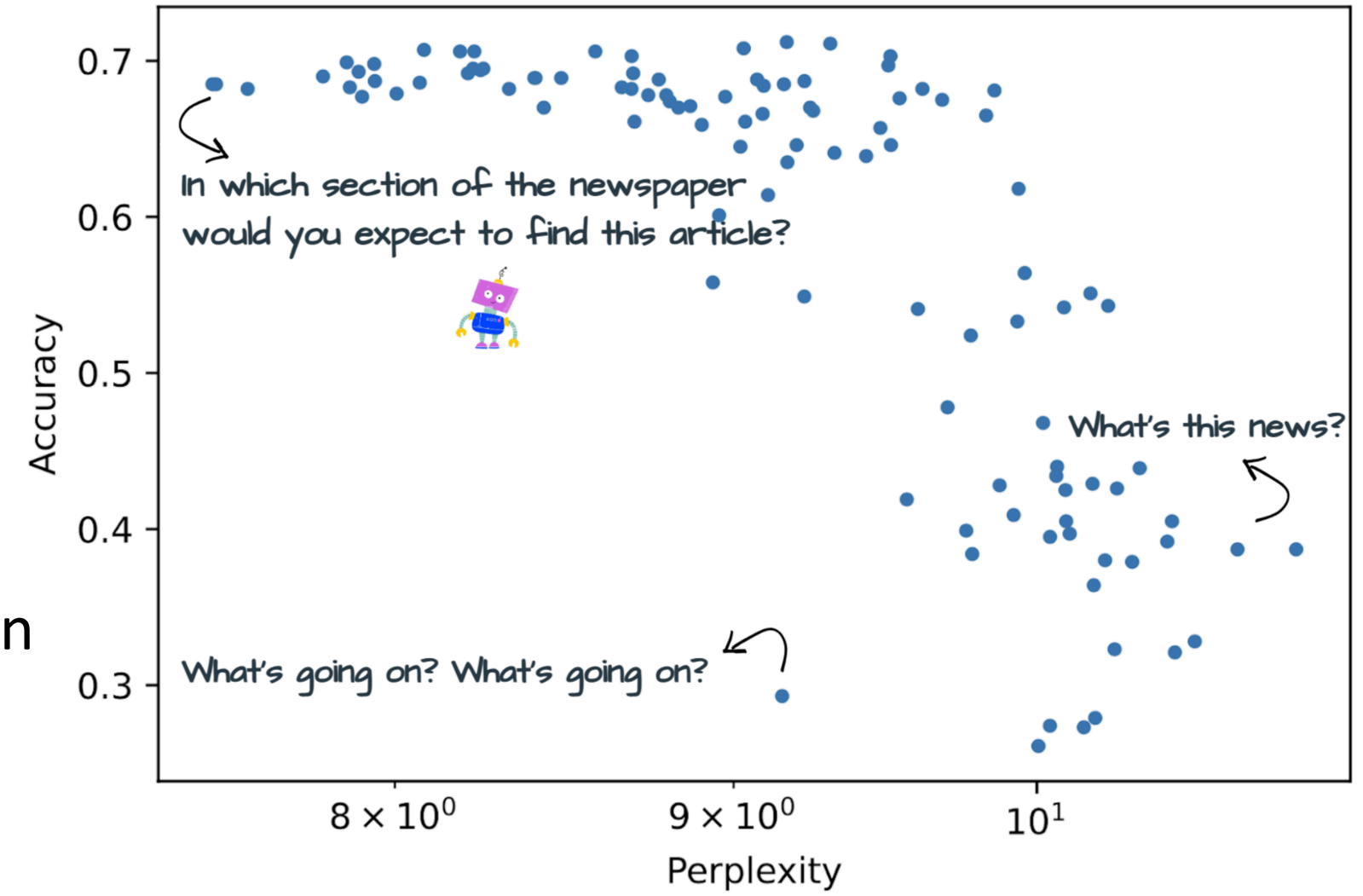
Li and Liang 2021; Lester et al. 2021

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



Gonen et al. (2022)

Variability in Prompts

Task	Perplexity-score corr. Pearson	Perplexity-score corr. Spearman	Perplexity-acc corr. Pearson	Perplexity-acc corr. Spearman	Avg Acc	Acc 50%
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: $\boldsymbol{v}(\boldsymbol{x}_1)\ \boldsymbol{v}(y_1)\ \boldsymbol{v}(\boldsymbol{x}_2)\ \boldsymbol{v}(y_2)\ \ldots\ \boldsymbol{v}(\boldsymbol{x}_{test})$

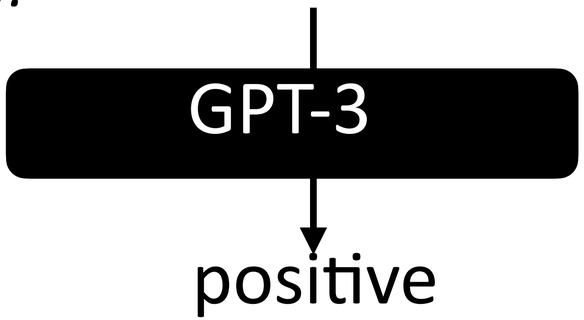
Review: The cinematography was stellar; great movie!
Sentiment (positive or negative): positive

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):



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.

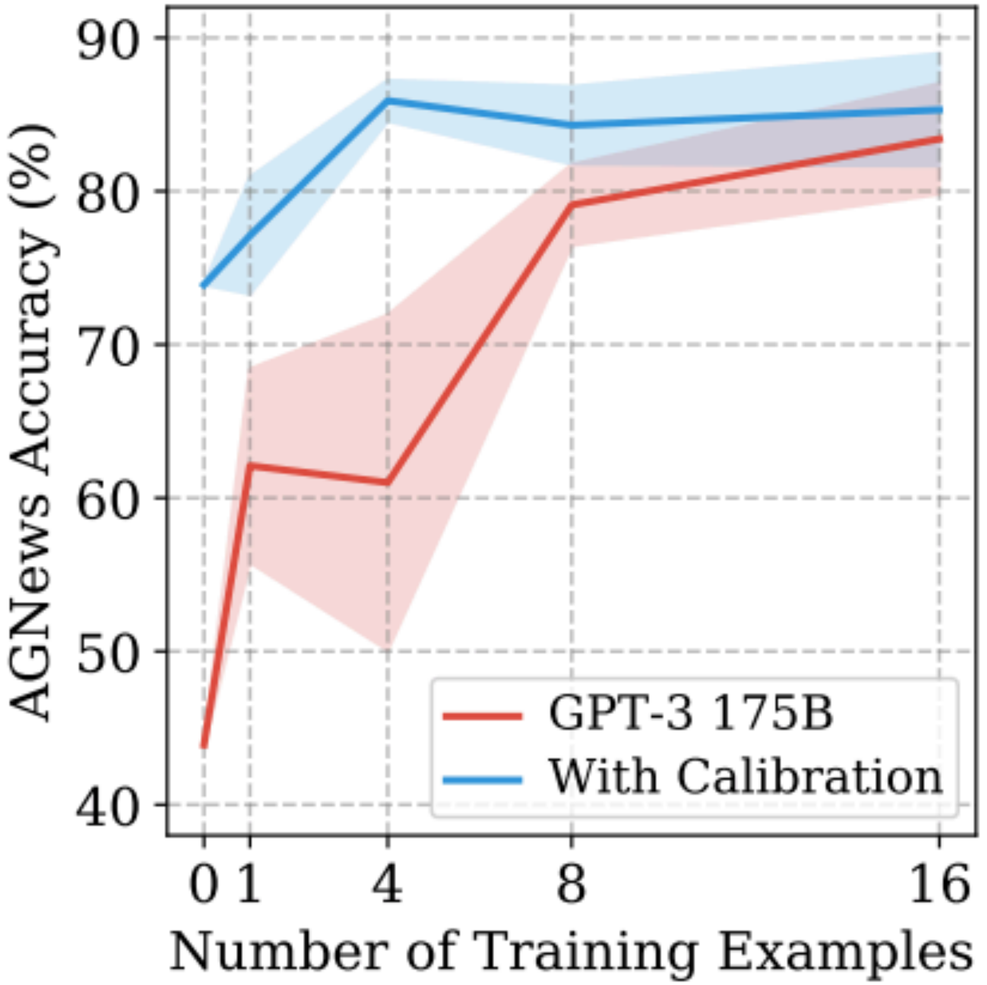
Sentiment: positive

Review: Okay movie but could've been better.

Sentiment: 

What can go wrong?

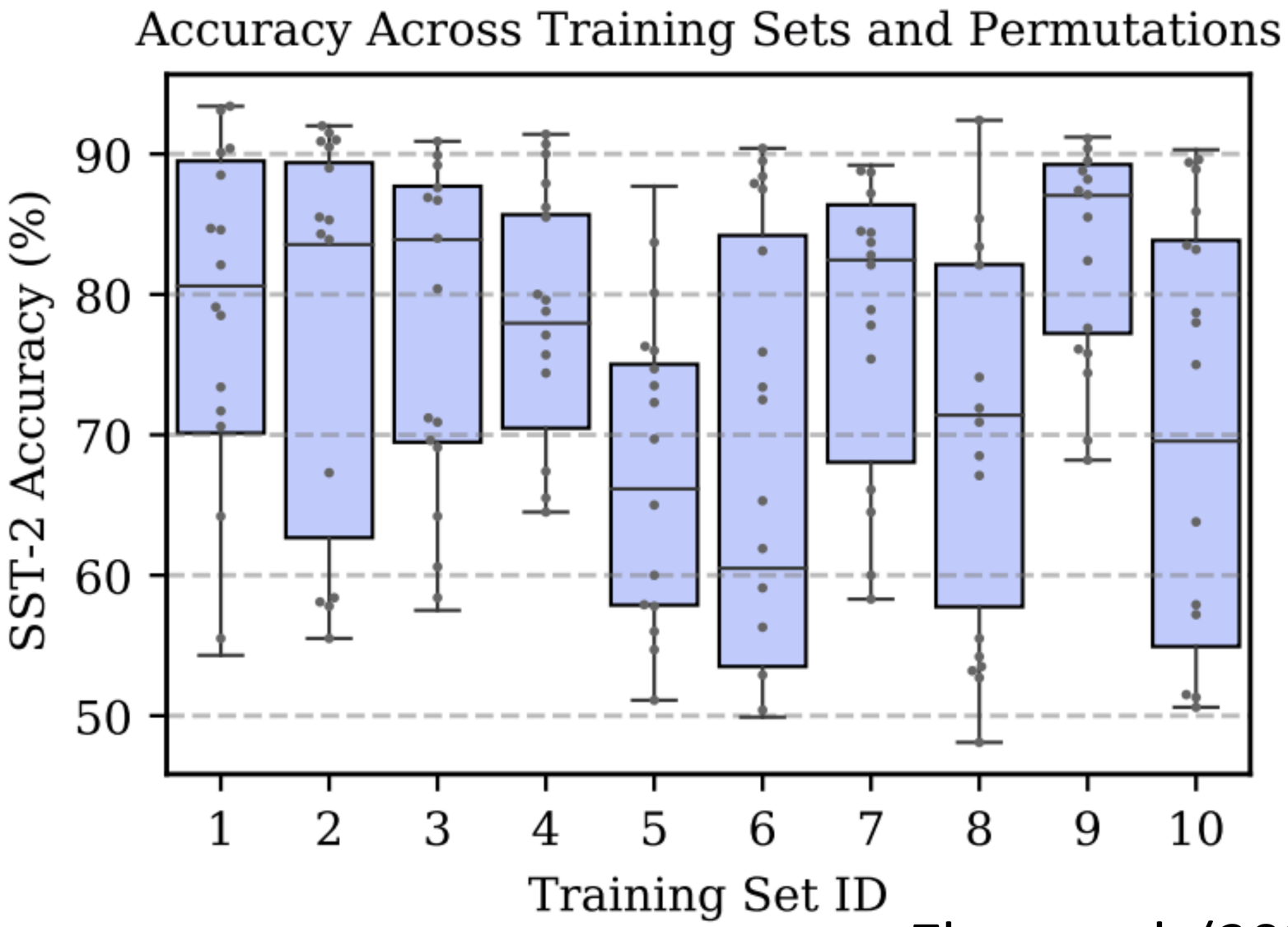
- All one training label — model sees extremely skewed distribution
- What if we take random sets of training examples? There is quite a bit of variance on basic classification tasks
- Note: these results are with basic GPT-3 and not Instruct-tuned versions of the model. This issue has gotten a lot better



Zhao et al. (2021)

What can go wrong?

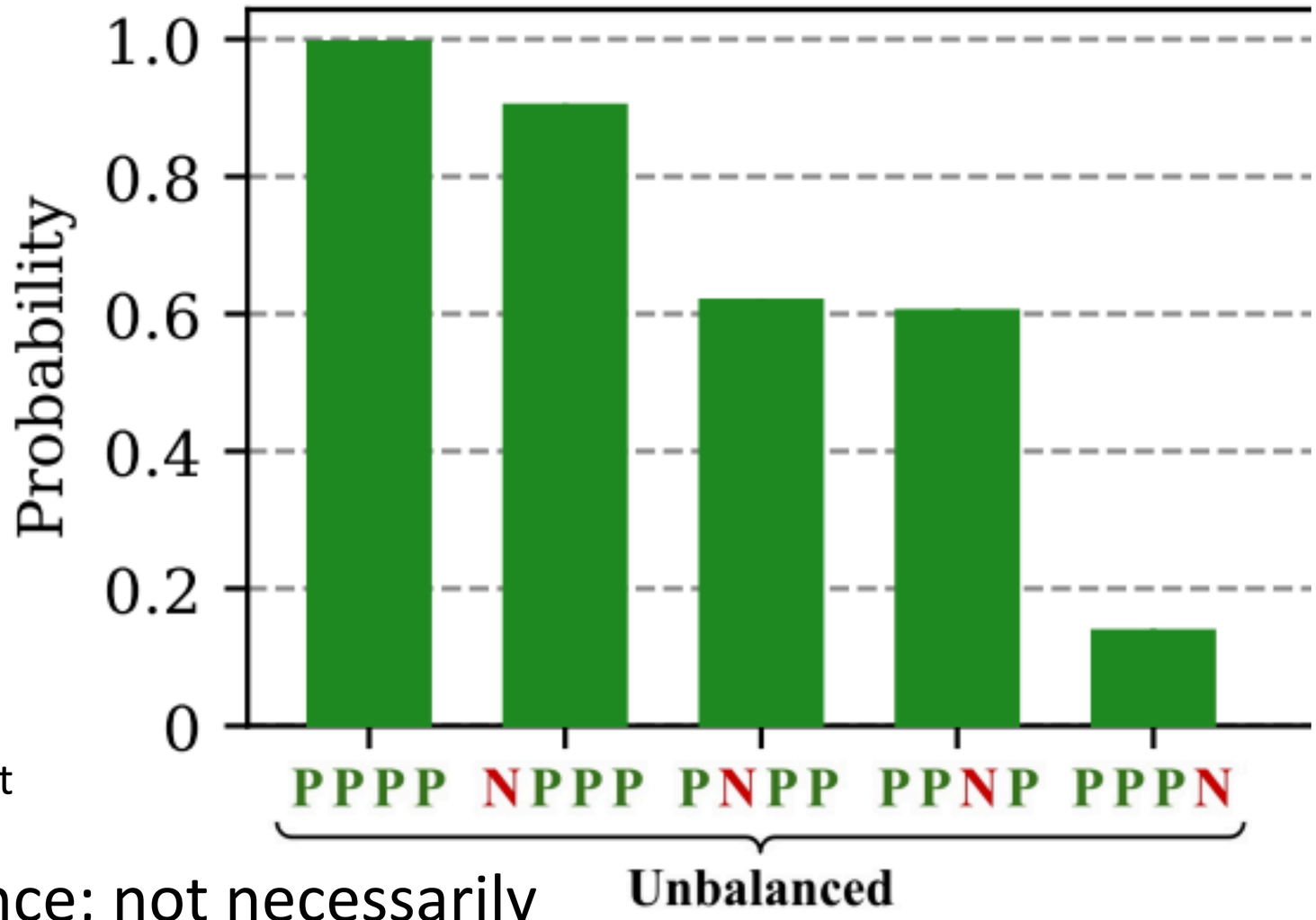
- Varies even across permutations of training examples
- x-axis: different collections of train examples.
  y-axis: sentiment accuracy. Boxes represent results over different permutations of the data



Zhao et al. (2021)

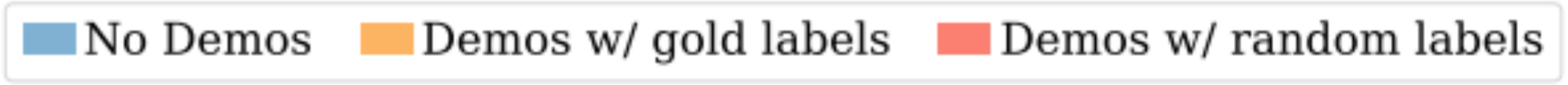
What can go wrong?

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

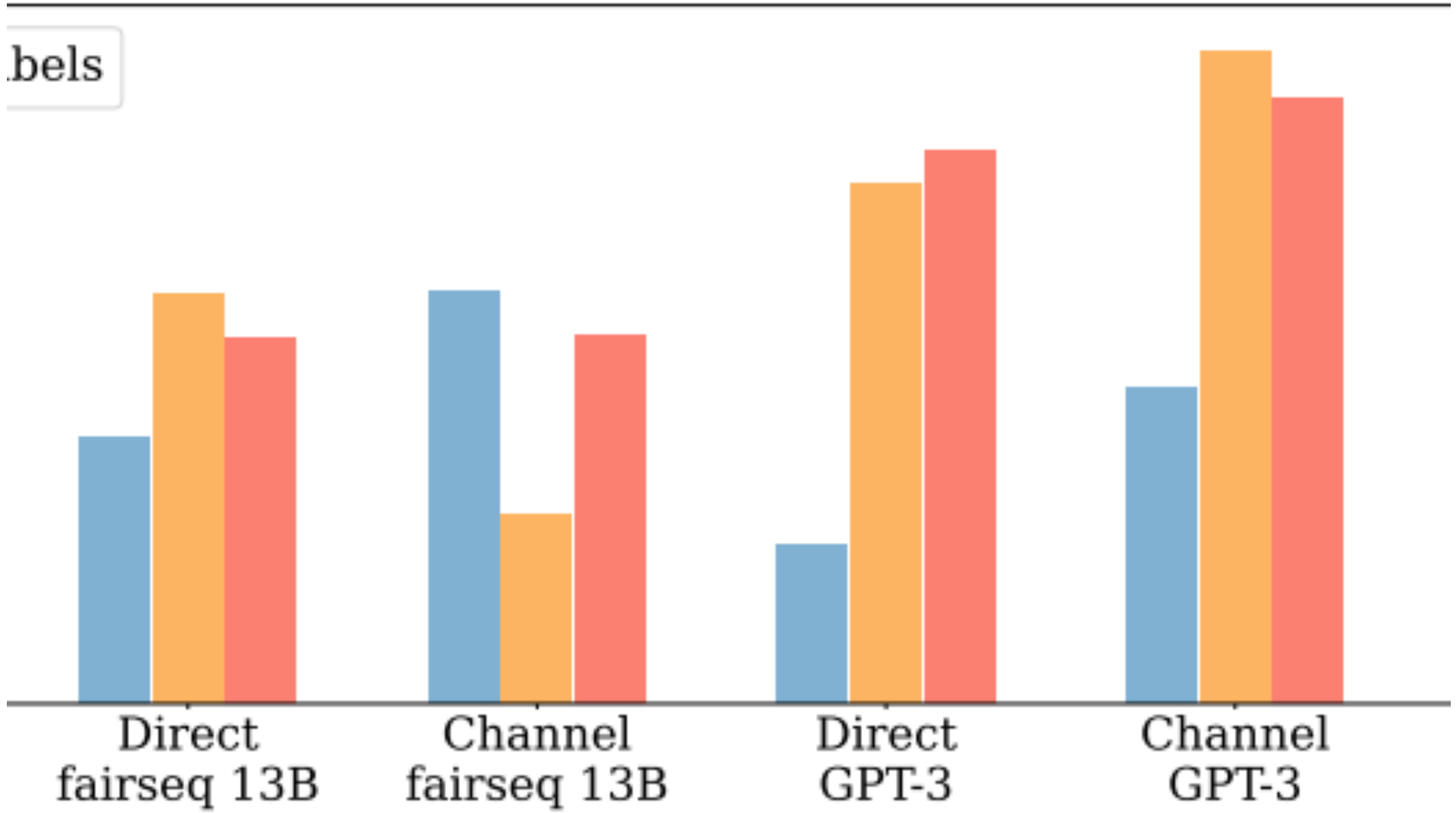


Zhao et al. (2021)

Rethinking Demonstrations

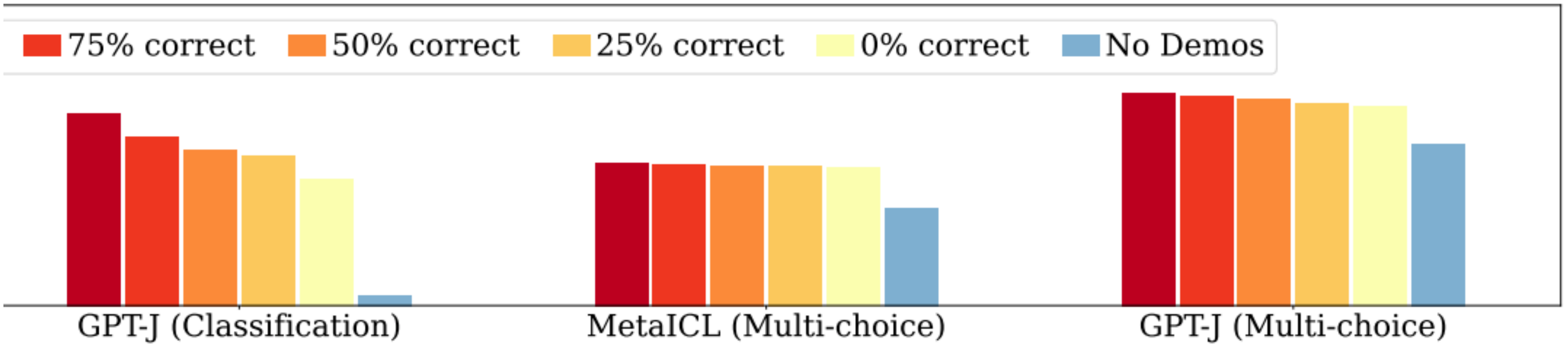


- Surprising result: how necessary even are the demonstrations?
- Using random labels does not substantially decrease performance??



Min et al. (2022)

Rethinking Demonstrations

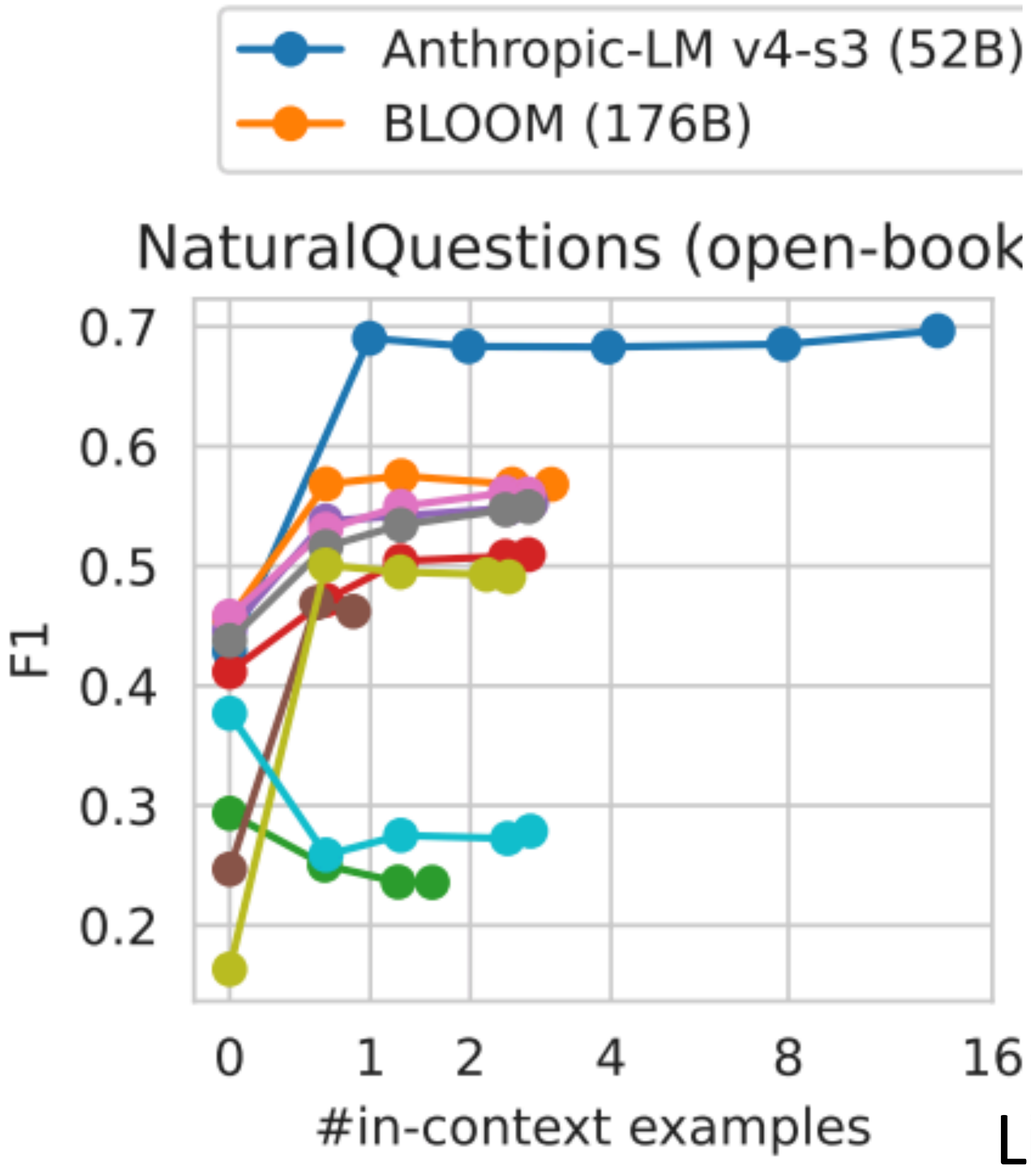


- Having even mislabeled demonstrations is much better than having no demonstrations, indicating that the form of the demonstrations is partially responsible for in-context learning

Min et al. (2022)

Results: HELM

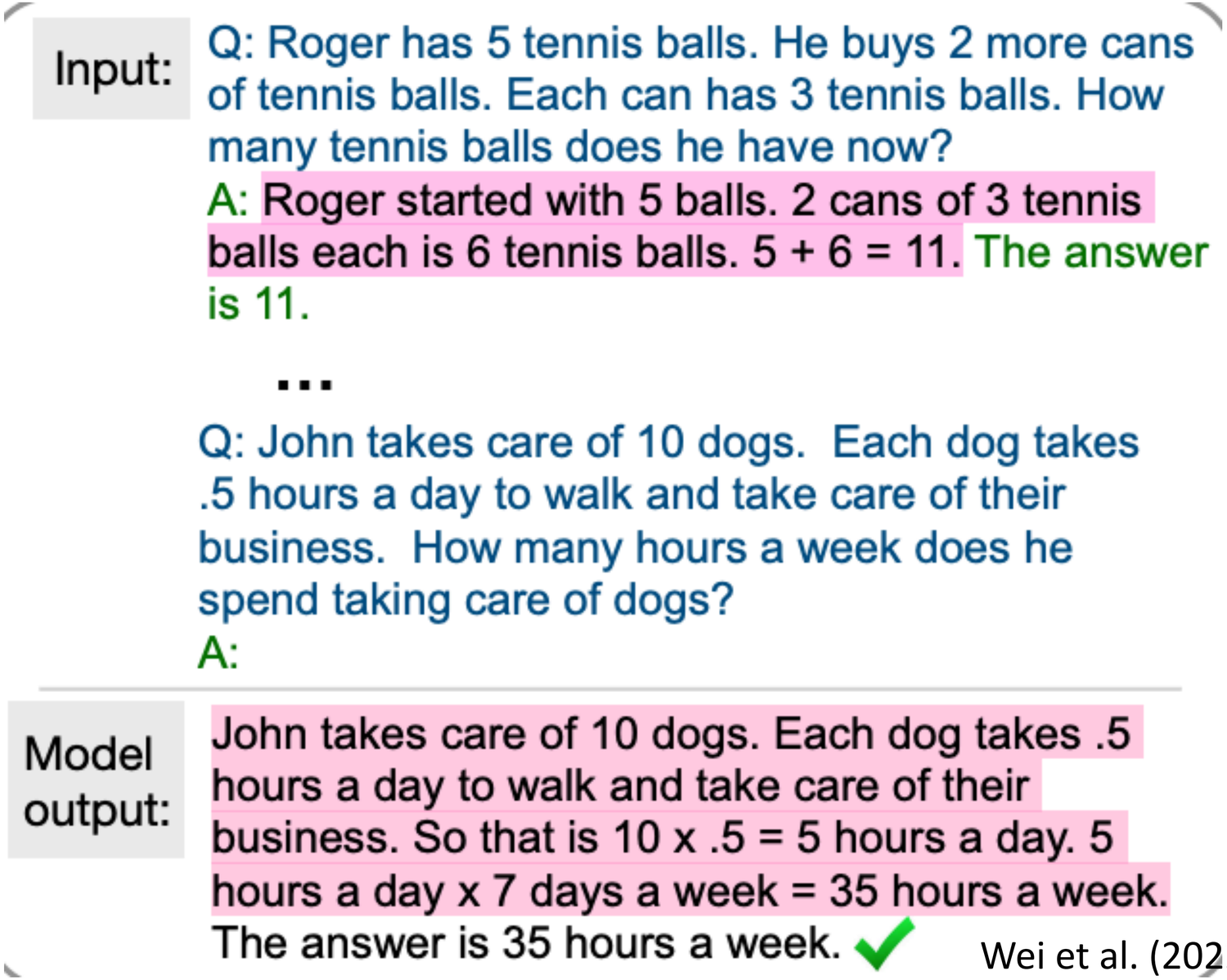
- So, how much better is few-shot compared to zero-shot?
- Each line is a different LM
- More in-context examples generally leads to better performance
- What do we see here?



Liang et al. (2022)

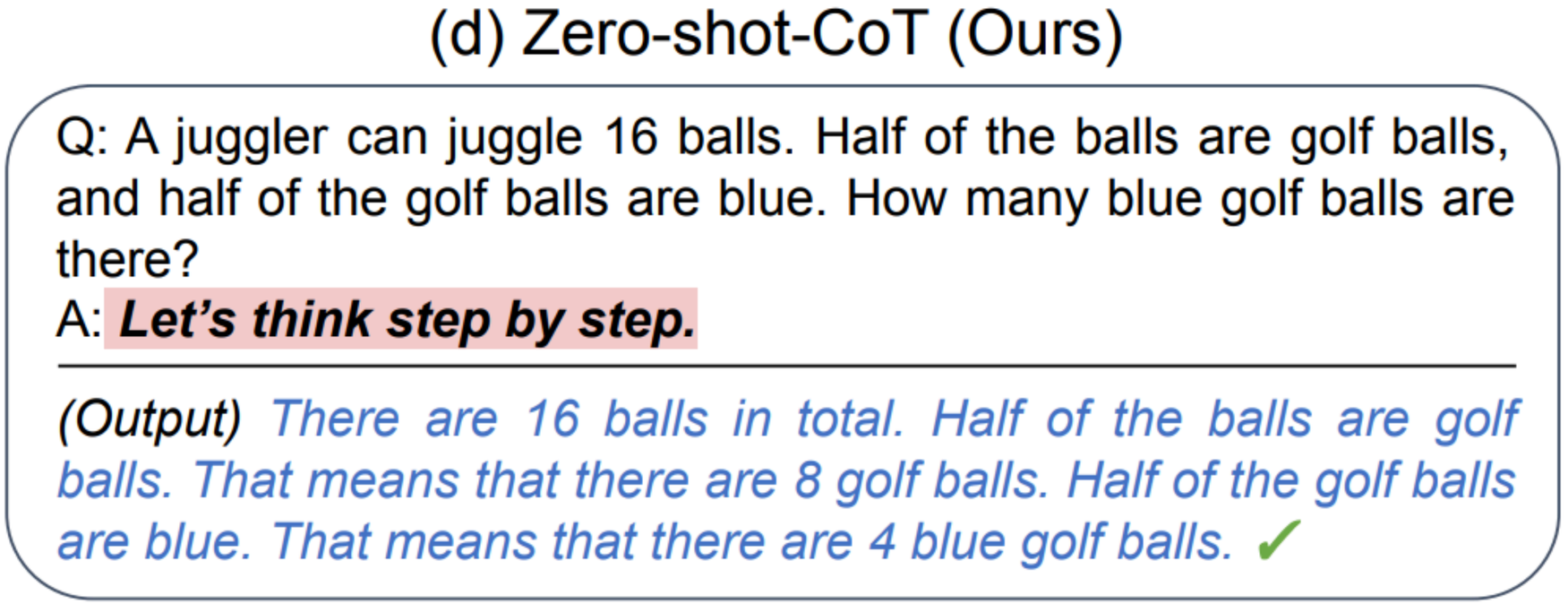
Chain-of-thought

- 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



Wei et al. (2022)

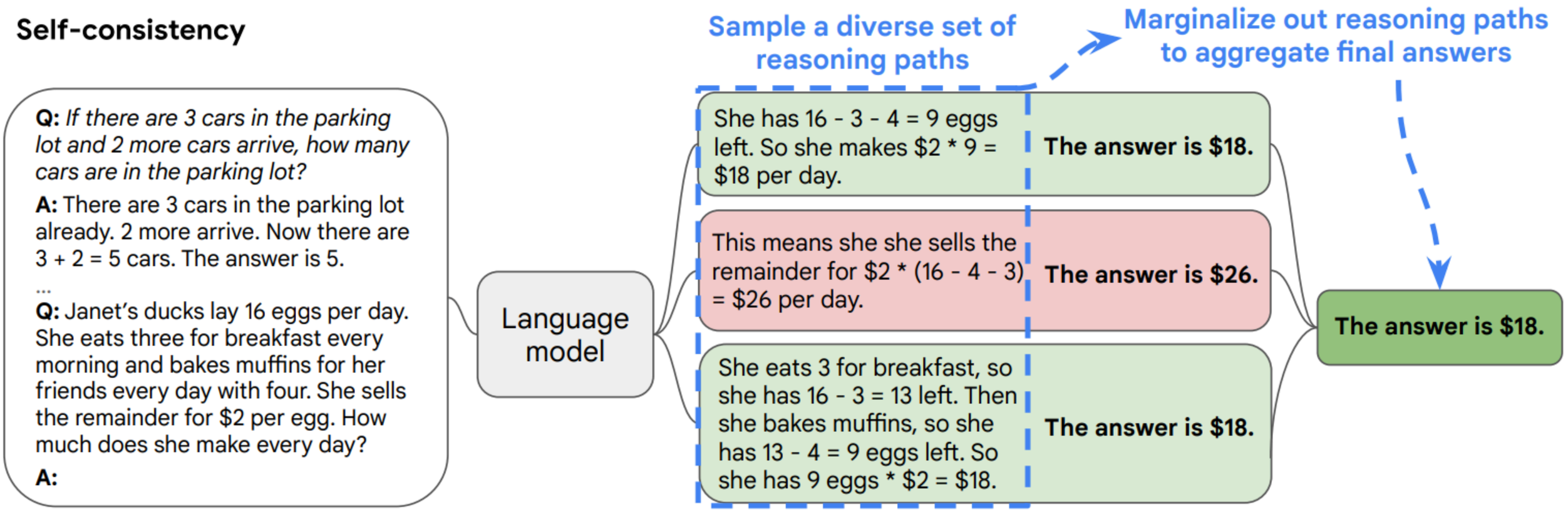
Step-by-Step



- Prompt for step-by-step reasoning: produces chains of thought without including demonstrations
- Separate prompt to extract the answer ("Therefore, the answer is ___")

Kojima et al. (2022)

Self-Consistency



- 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

Wang et al. (2022)

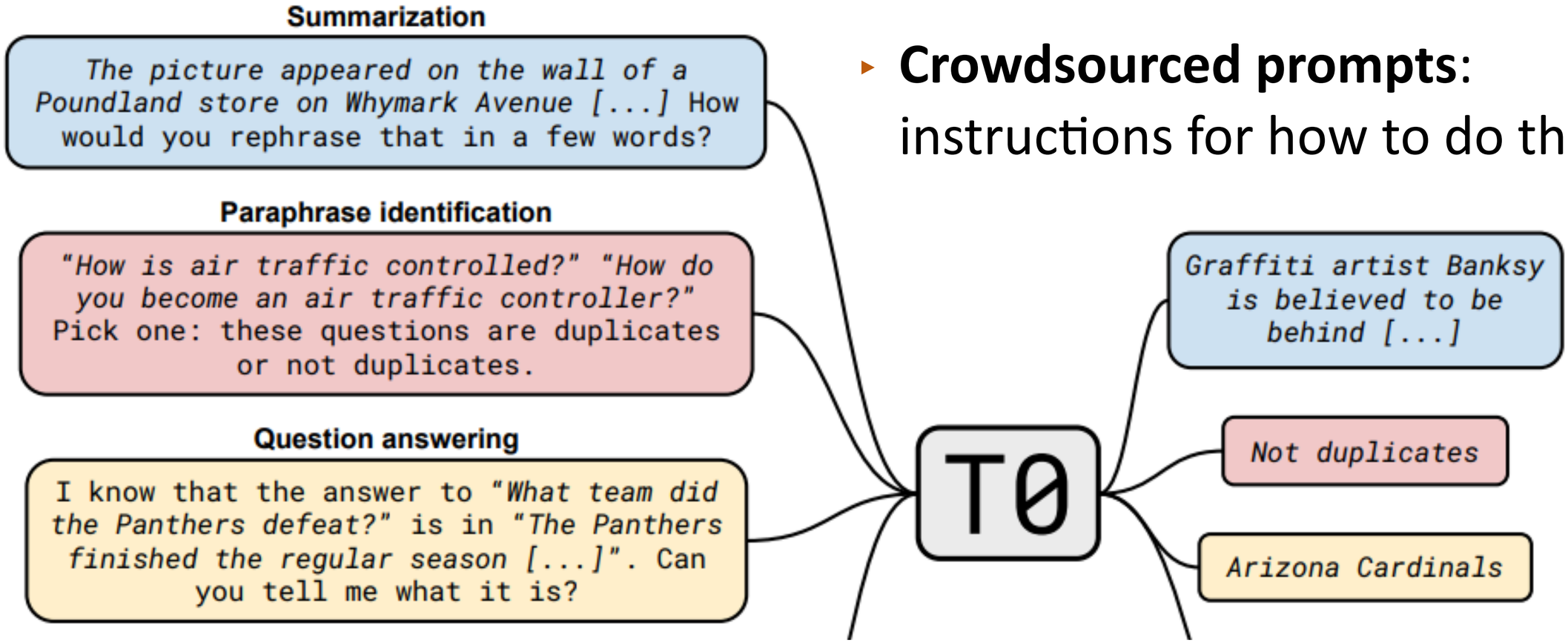
Multi-Task Model Engineering: Instruction Tuning

Instruction Tuning

- We want to optimize models for P(answer | 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

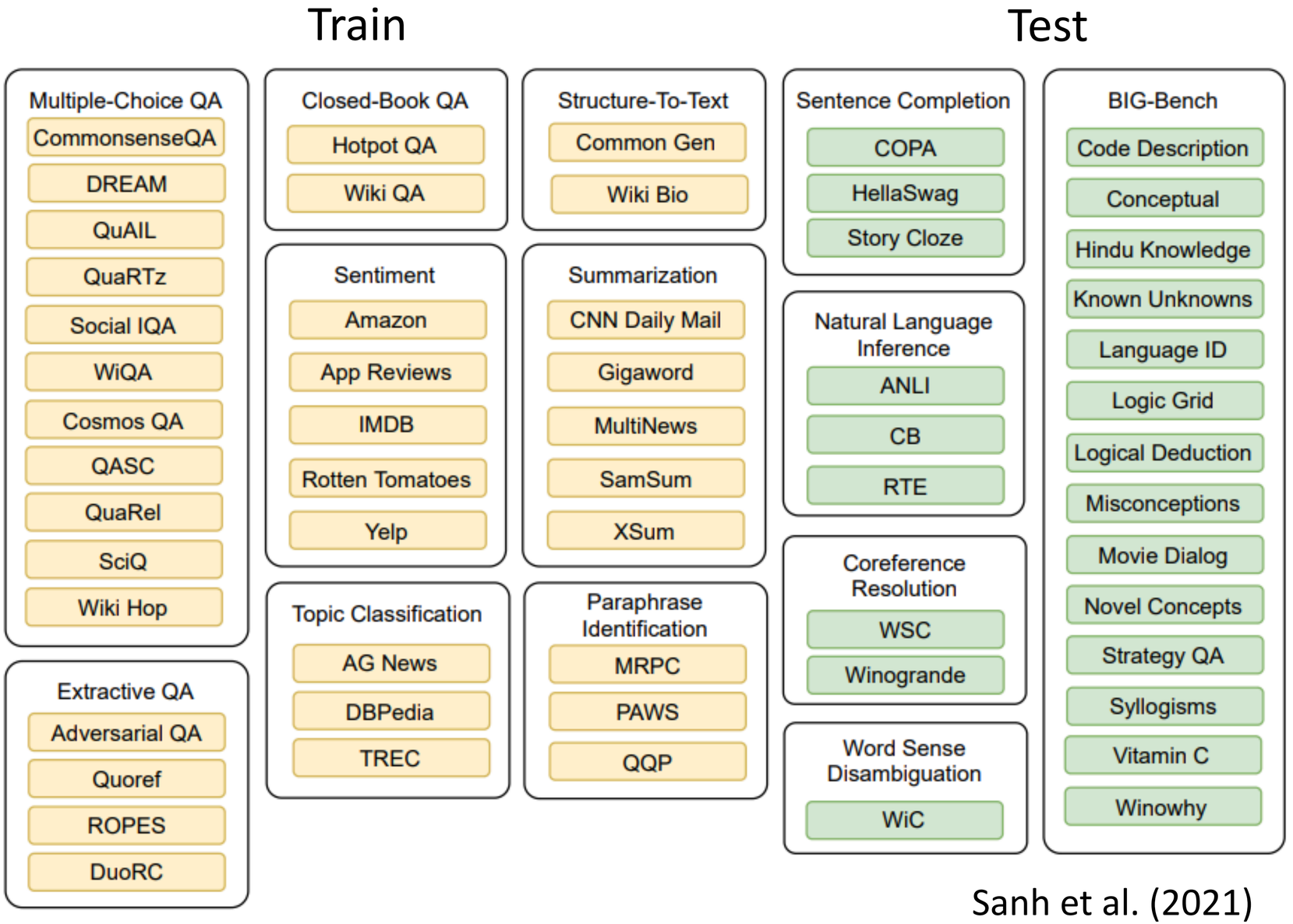
- T0: tries to deliver on the goal of T5 and do many tasks with one model
- **Crowdsourced prompts**: instructions for how to do the tasks



Sanh et al. (2021)

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**



Sanh et al. (2021)

Flan-T5

- Flan-T5: T5-11B model given the “Flan treatment”, instruction tuned on many tasks
- Best model at the ~10B scale for few-shot prompting, also very good choice for fine-tuning

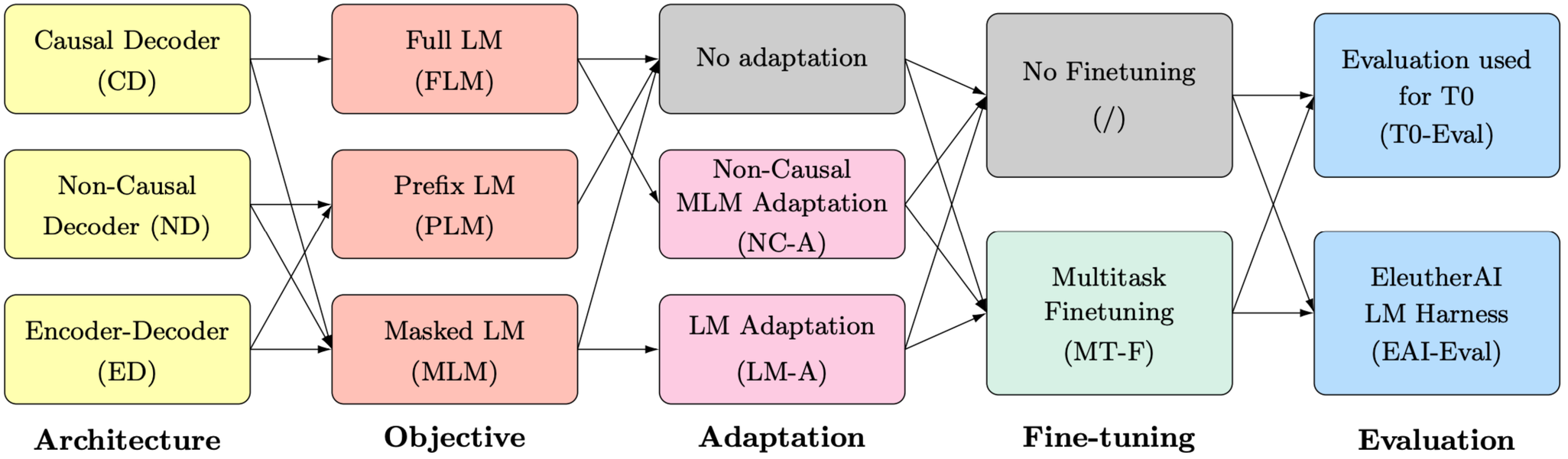
			MMLU		BBH	
Params	Model	Norm. avg.	Direct	CoT	Direct	CoT
11B	T5-XXL	-2.9	25.9	18.7	29.5	19.3
	Flan-T5-XXL	23.7 **(+26.6)**	55.1	48.6	45.3	41.4

(about 20% behind the 540B Flan-PaLM)

- If you have the resources, Flan-T5 is something you can explore in your project/ research!

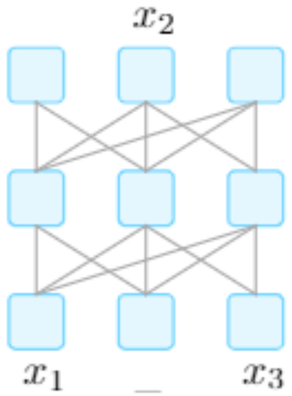
Chung et al. (2022)

Training Zoo

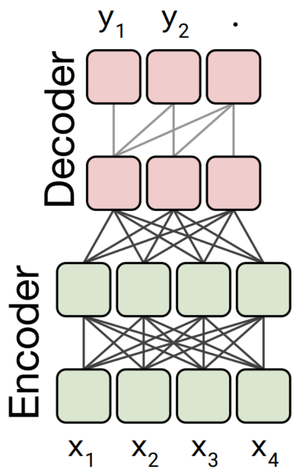


Wang et al. 2022

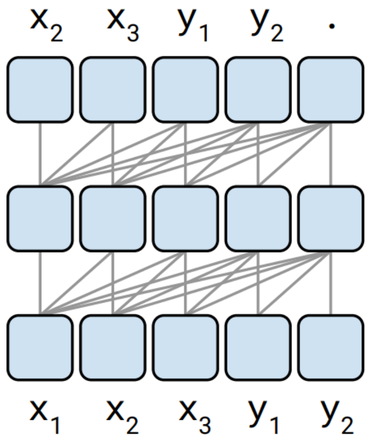
Model Zoo



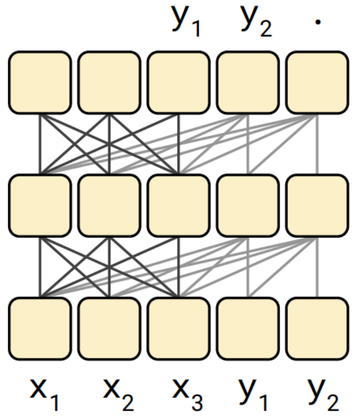
Masked Language Model
(e.g. BERT)



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

Questions?