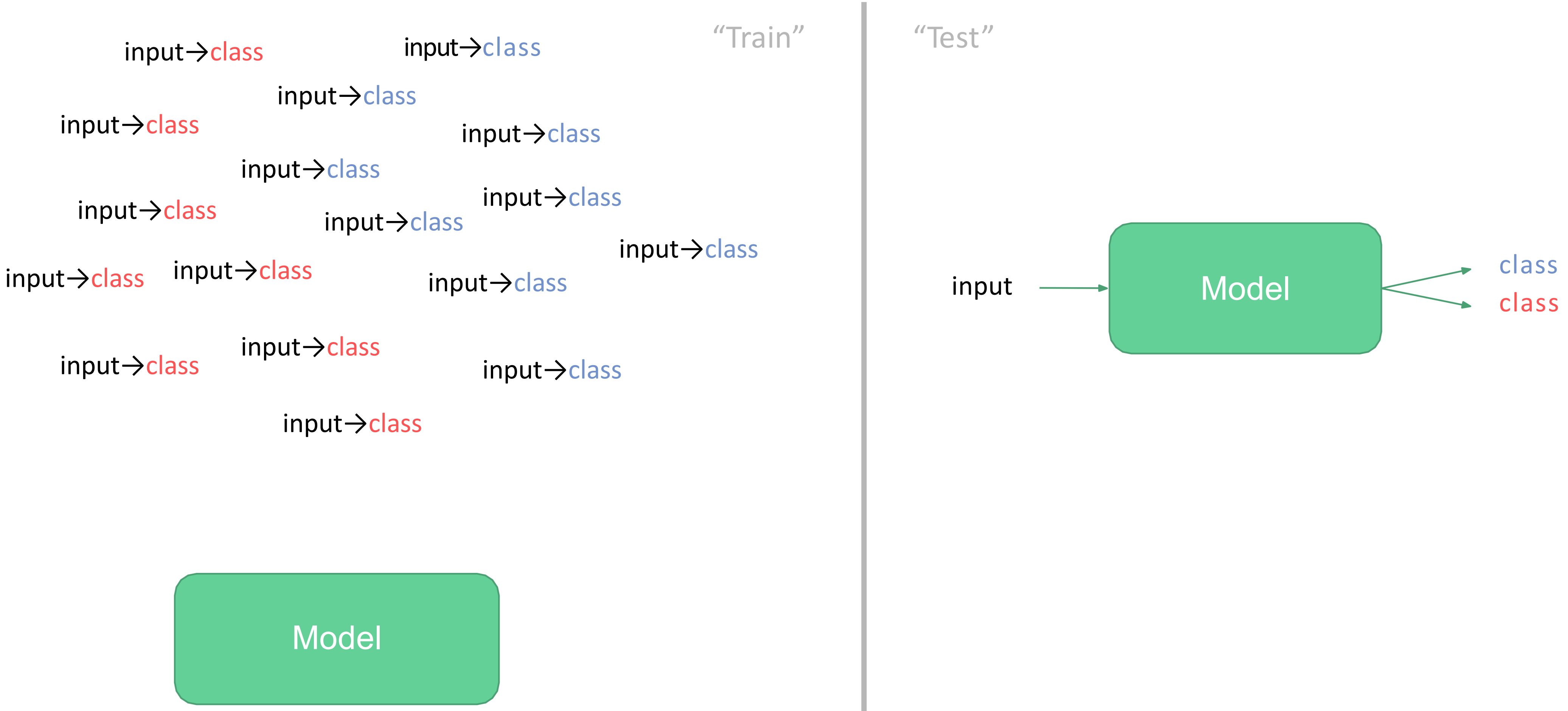
Prompting

https://github.com/allenai/acl2022-zerofewshot-tutorial

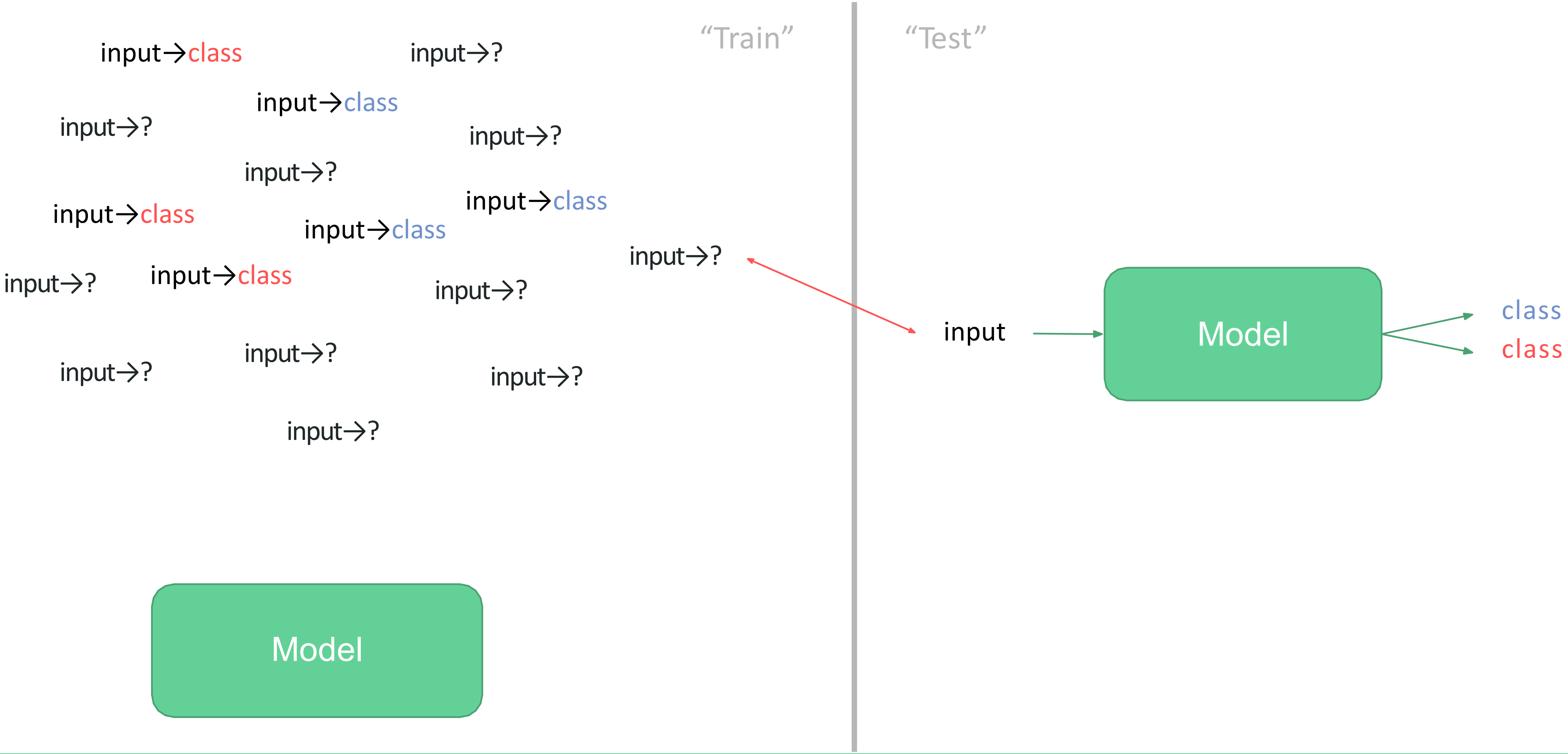
https://arxiv.org/pdf/2107.13586.pdf

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

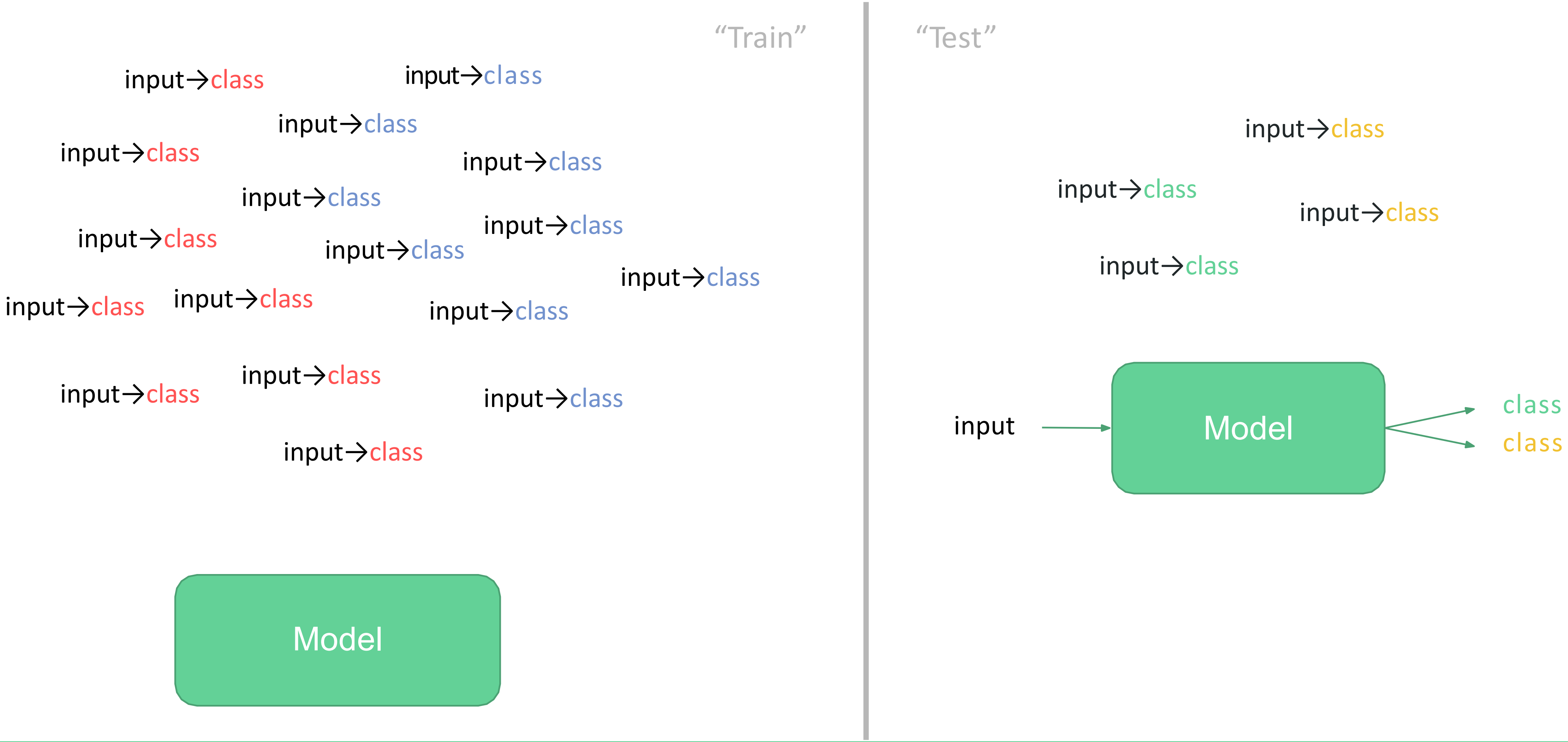
Related Ideas: Supervised Learning



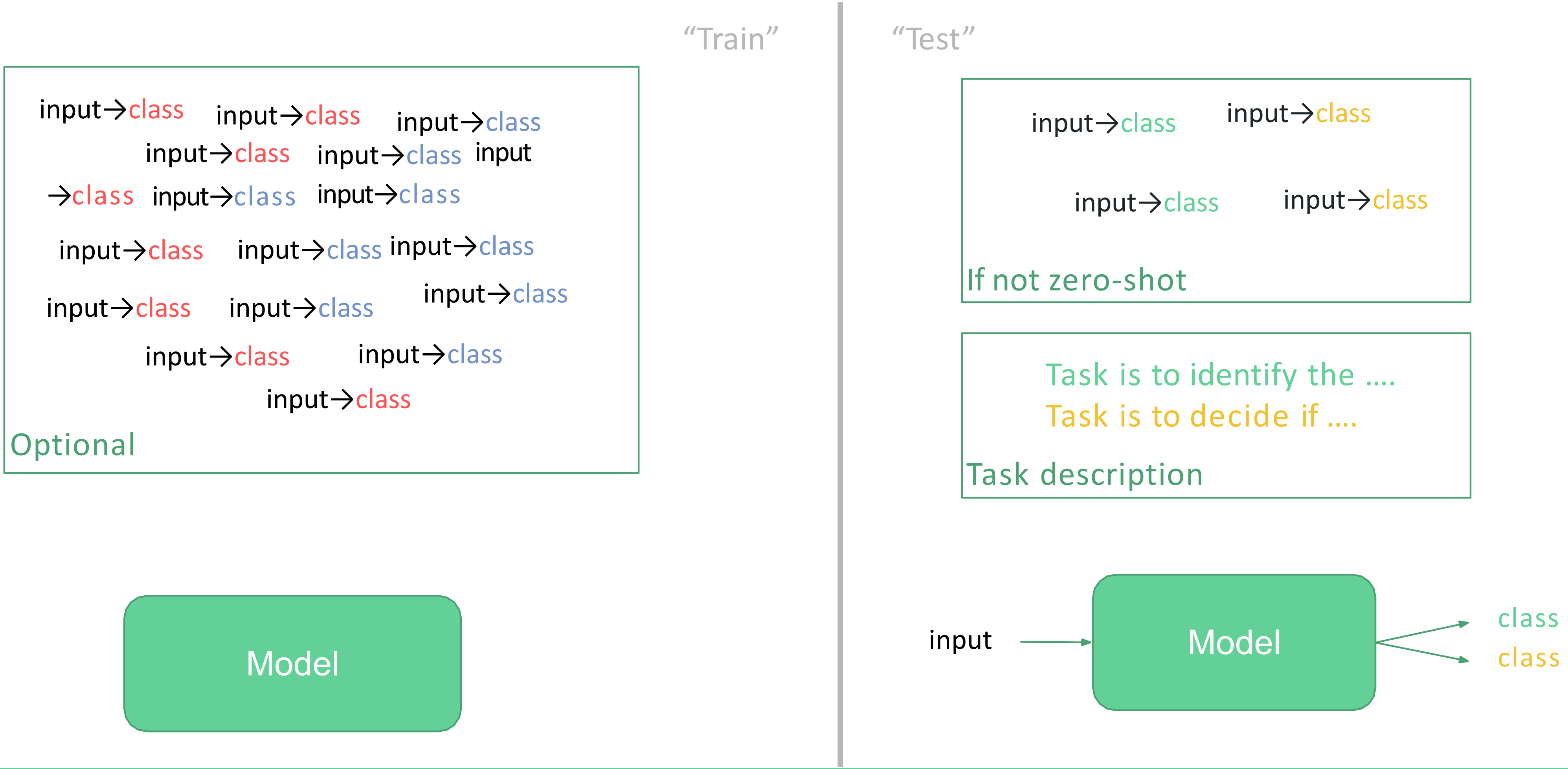
Related Ideas: Semi-Supervised Learning



Related Ideas: (traditional) Few-shot learning



Related Ideas: (modern) Few-shot learning



N-way-*K*-shot Classification

Class 1	Instance 1	Instance 2	Instance 3	…	Instance K
Class 2	Instance 1	Instance 2	Instance 3	…	Instance K
Class 3	Instance 1	Instance 2	Instance 3	…	Instance K
…	…	…	…		…
Class N	Instance 1	Instance 2	Instance 3	…	Instance K

Often tough in NLP

- Imbalanced classes
  - Can’t control distribution
- Open ended classes
  - E.g. topics
- Select from a context
  - E.g. QA
- Text generation
  - E.g. summarization

For the most part, we will use *K* for total labeled examples

LM Prompting

A three-hour cinema master class.

positive

negative

LM Prompting

LM Head

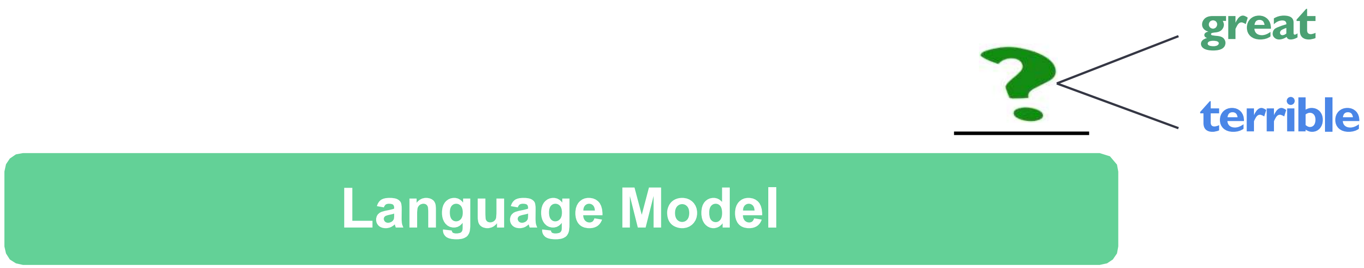
(Finetuned) Language Model

[CLS] A three-hour cinema master class. It was

Perform the task without finetuning?

LM Prompting

(Frozen)



A three-hour cinema master class. It was

P1 = P(It was **great**! | A three-hour cinema master class.)

P2 = P(It was **terrible**! | A three-hour cinema master class.)

P1>P2 "**positive**"

P1<P2 "**negative**"

Brown et al. 2020. "Language Models are Few-Shot Learners"

In-context Learning (GPT3; Brown et al., 2020)

Movie review dataset

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

An effortlessly accomplished and richly resonant work. It was great!
A mostly tired retread of several other mob tales. It was terrible!
A three-hour cinema master class. It was ________

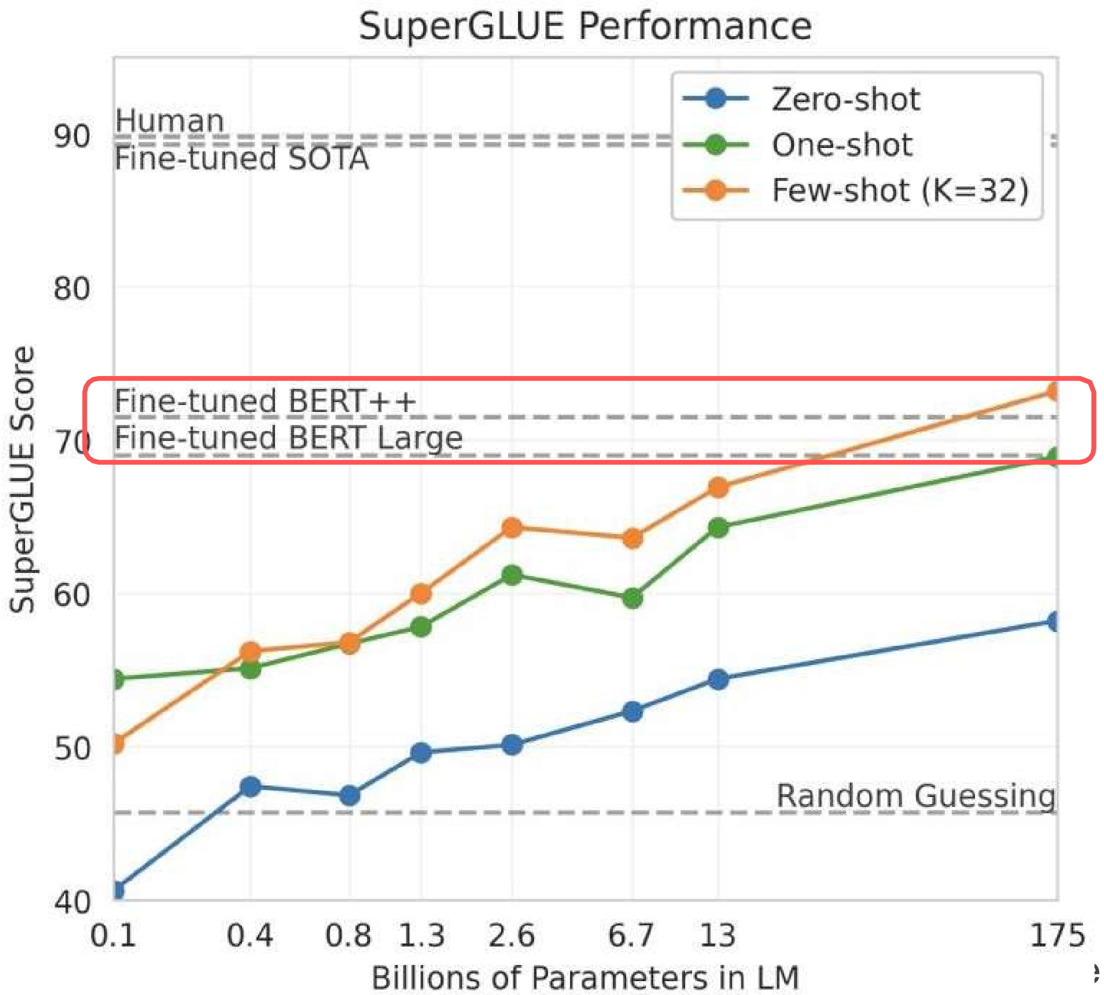
Language Model

P1 = P(It was **great**! | 1st train input+output \n 2nd train input+output \n A three-hour cinema master class.)

P2 = P(It was **terrible**! | 1st train input+output \n 2nd train input+output \n A three-hour cinema master class.)

P1>P2 "**positive**"
P1<P2 "**negative**"

In-context learning results



: Models are Few-Shot Learners”

Terminologies

Input to the LM

An effortlessly accomplished and richly resonant work.	It was great!
A mostly tired retread of several other mob tales.	It was terrible!
A three-hour cinema master class.	It was ______!

***Prompt*:** A conditioning text coming before the test input

Demonstrations: A special instance of prompt which is a concatenation of the k-shot training data (in in-context learning, prompt==demonstrations)

Terminologies

Input to the LM

An effortlessly accomplished and richly resonant work. It was great!

A mostly tired retread of several other mob tales. It was terrible!

A three-hour cinema master class. It was ______!

Prompt: A conditioning text coming before the test input

Demonstrations: A special instance of prompt which is a concatenation of the k-shot training data (in in-context learning, prompt==demonstrations)

Pattern: A function that maps an input to the text (a.k.a. template)

Verbalizer: A function that maps a label to the text (a.k.a. label words)

Examples of patterns/verbalizers

An effortlessly accomplished and richly resonant work. It was great!

A mostly tired retread of several other mob tales. It was terrible!

A three-hour cinema master class. It was great!

Pattern: f(<x>) = <x>
Verbalizer: v("positive") = "It was great!", f("negative") = "It was terrible!"

Review: An effortlessly accomplished and richly resonant work. Sentiment: positive

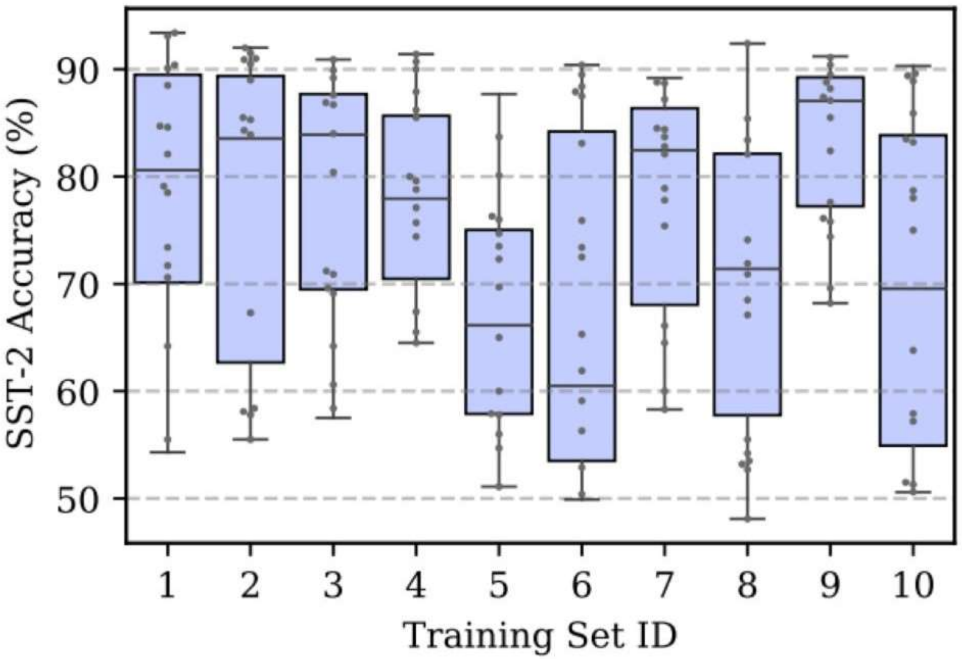
Review: A mostly tired retread of several other mob tales. Sentiment: negative

Review: A three-hour cinema master class. Sentiment: positive

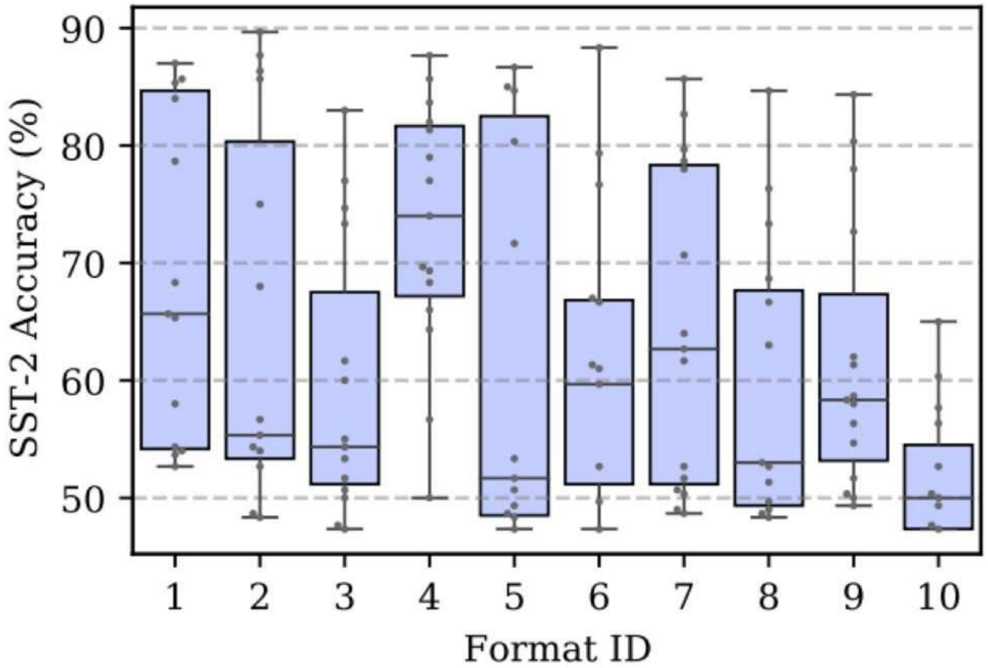
Pattern: f(<x>) = "Review: <x>"
Verbalizer: v(<x>) = "Sentiment: <x>"

Variance

Across different training sets and permutations



Across different training sets and patterns/verbalizers



Zhao et al. 2021. "Calibrate Before Use: Improving Few-Shot Performance of Language Models"

Variance

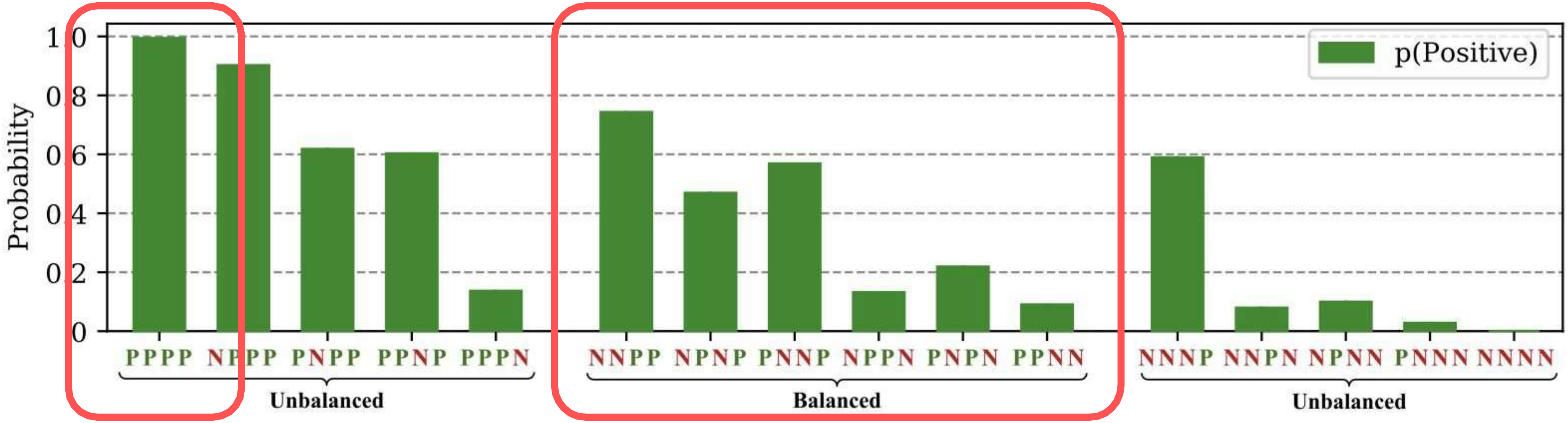


Figure 4. **Majority label and recency biases** cause GPT-3 to become biased towards certain answers and help to explain the high variance across different examples and orderings. Above, we use 4-shot SST-2 with prompts that have different class balances and permutations, e.g., [P P N N] indicates two positive training examples and then two negative. We plot how often GPT-3 2.7B predicts Positive on the balanced validation set. When the prompt is unbalanced, the predictions are unbalanced (*majority label bias*). In addition, balanced prompts that have one class repeated near the end, e.g., end with two Negative examples, will have a bias towards that class (*recency bias*).

Zhao et al. 2021. "Calibrate Before Use: Improving Few-Shot Performance of Language Models"

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

Input: A three-hour master class.
Label: ________

Language Model

Input: An effortlessly accomplished and richly resonant work.
Label: **negative**

Input: A mostly tired retread of several other mob tales.
Label: **positive**

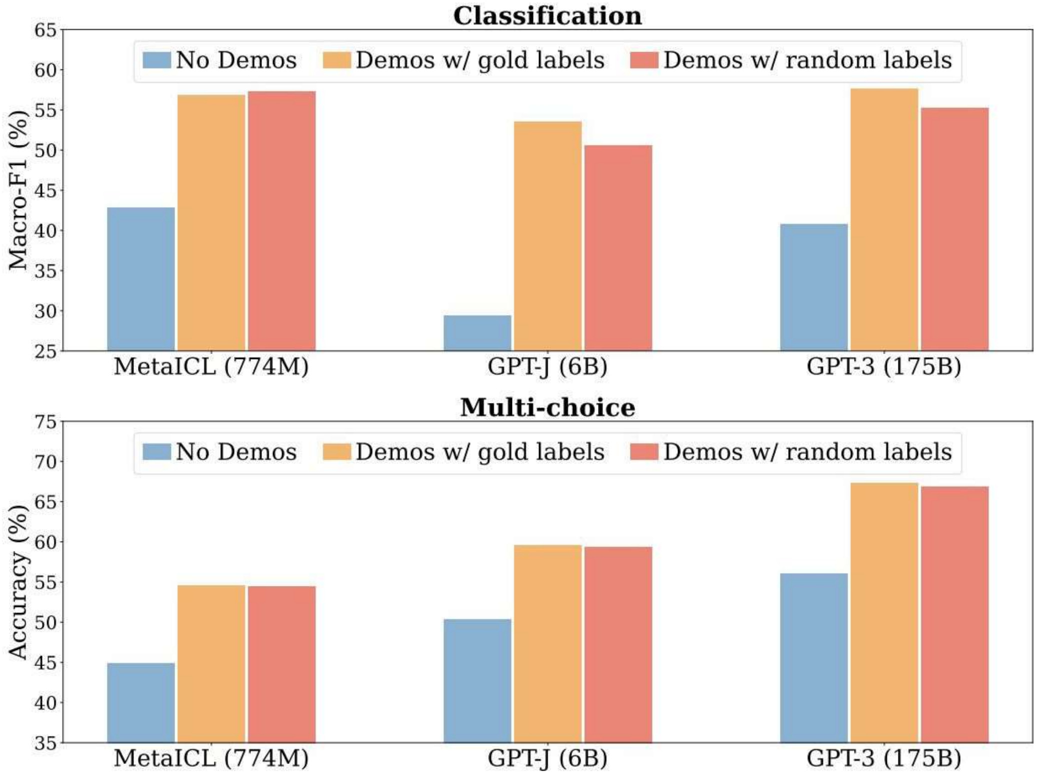
Input: A three-hour master class.
Label: ________

Language Model

Min et al. 2022. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?"

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping



Min et al. 2022. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?"

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping

Input: An effortlessly accomplished and richly resonant work.
Label: positive

Input: A mostly tired retread of several other mob tales.
Label: negative

Input: A three-hour master class.
Label: ________

Language Model

Min et al. 2022. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?"

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping

Input: Colour-printed lithograph.Very good condition.
Label: positive

Input: Many accompanying marketing …meaning.
Label: negative

Input: A three-hour master class.
Label: ________

Language Model

Removing correct input distribution significantly drops performance

Min et al. 2022. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?"

Impact of input-label mapping

In-context learning does not necessitate correct input-label mapping

Input: An effortlessly accomplished and richly resonant work.
Label: Unanimity

Input: A mostly tired retread of several other mob tales.
Label: Wave

Input: A three-hour master class.
Label: ________

Language Model

Removing correct input distribution significantly drops performance

Removing correct label space significantly drops performance

Input and label distributions matter *independently*

Min et al. 2022. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?"

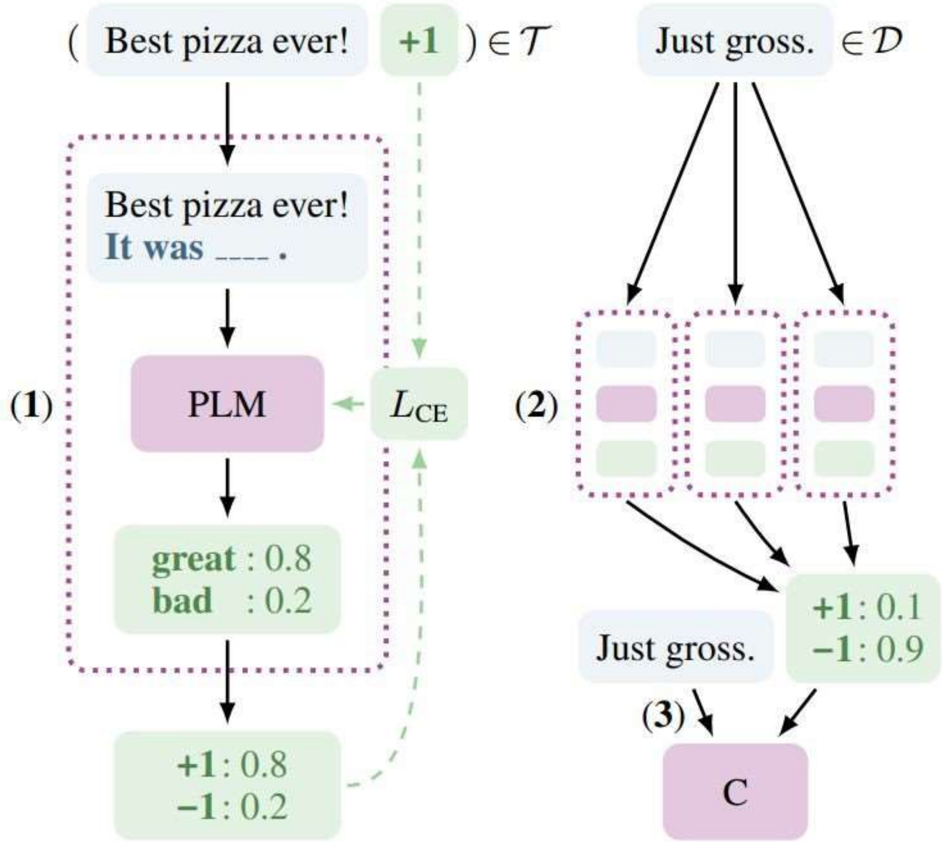
Summary & Open questions

- In-context learning has been a promising few-shot learning approach
  - No need for gradient updates →Much easier to use large models! (Even compared to parameter-efficient tuning covered in Section 3)
- Better calibration, better scoring of model outputs, better formation of demonstrations lead to great improvements
  - How to make it less sensitive?
  - It increases inference cost – how to make it efficient?
  - How to scale it (longer context, more training examples, wider range of tasks)?
- Need to be cautious in evaluation
- Still in progress on understanding how/why it works, with papers showing that in-context learning is about *task location* rather than learning a *new* task
  - Can we predict whether in-context learning would work on a given task or not?

Full Finetuning Approaches

Pattern exploiting training (PET)

1. Train an ensemble of classifiers using prompt-based finetuning in few-shot setting.
2. Collect weak labels for a pool of unlabeled data.
3. Use to train final classifier w/ traditional finetuning.



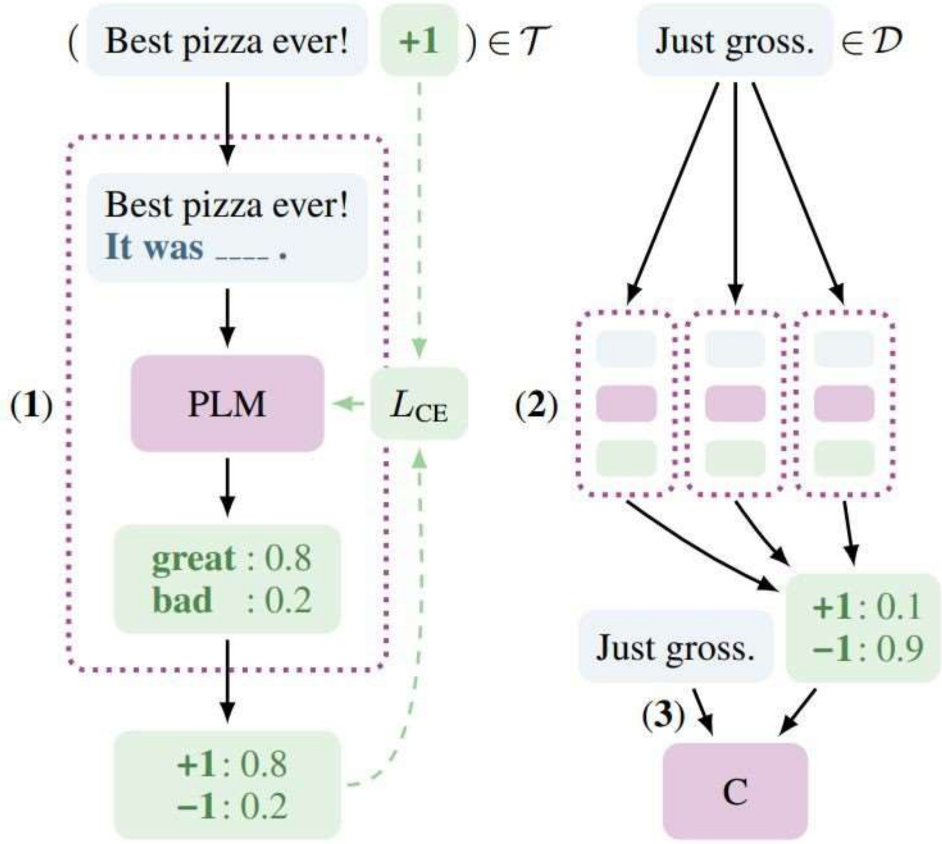
Schick and Schütze, 2020. "Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference"

Pattern exploiting training (PET)

Iterative PET (iPET)

Train several generations of PET models on datasets of increasing size.

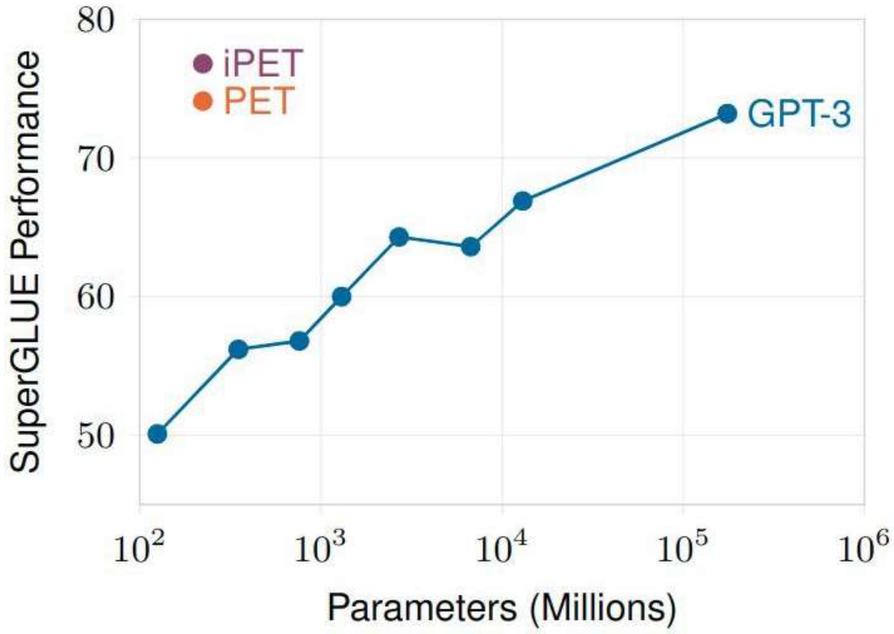
- Use output of other models to obtain labels.
- Select examples:
  - That models are more confident on.
  - That maintain label balance.



Schick and Schütze, 2020. "Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference"

PET - Results

- PET outperforms GPT-3 while using 1000x less parameters.
- Distillation approach consistently improves prompt-based finetuning.



Schick and Schütze, 2020. "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners"

Parameter-Efficient Finetuning

Parameter-Efficient Finetuning

	Model Size	Task-Specific Parameters
In-Context Learning	10B - 100B	Effectively None
Prompt-Based Finetuning	100M - 1B	All
Parameter-Eff. Finetuning	100M - 1B	<1% of model parameters

Parameter-Efficient Finetuning

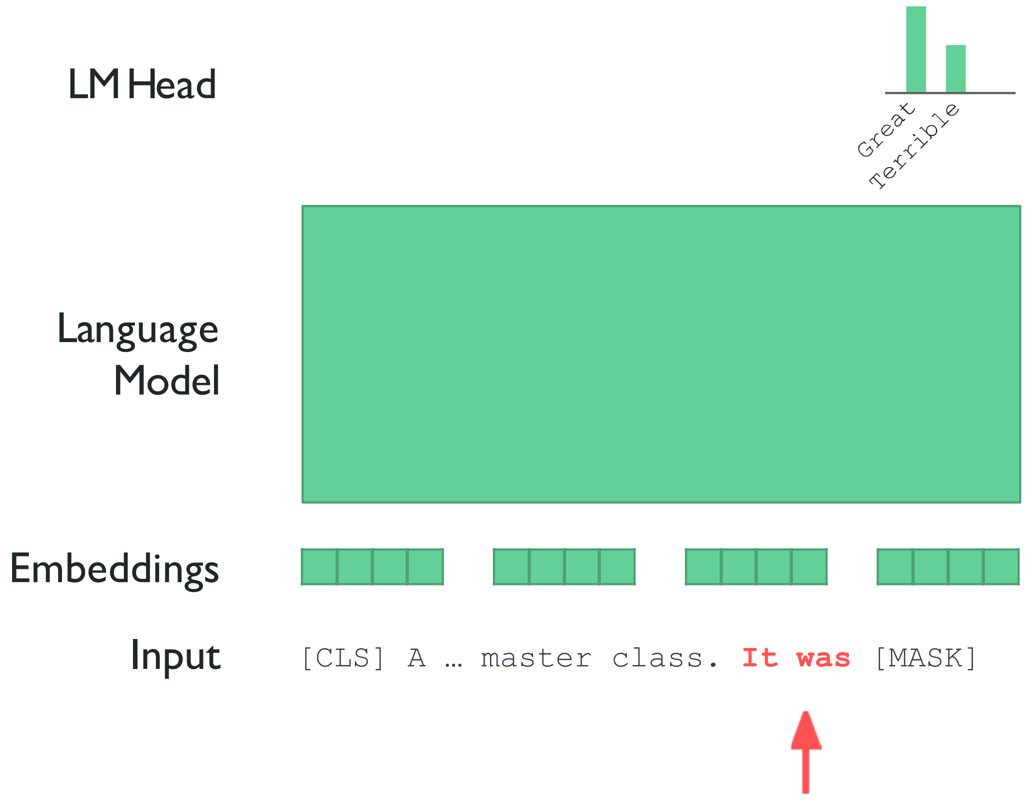
	Model Size	Task-Specific Parameters
In-Context Learning	10B - 100B	Effectively None
Prompt-Based Finetuning	100M - 1B	All
Parameter-Eff. Finetuning	100M - 1B	<1% of model parameters

Methods described in order of increasing competitiveness with prompt-based finetuning.

Input-level modifications

Two types:

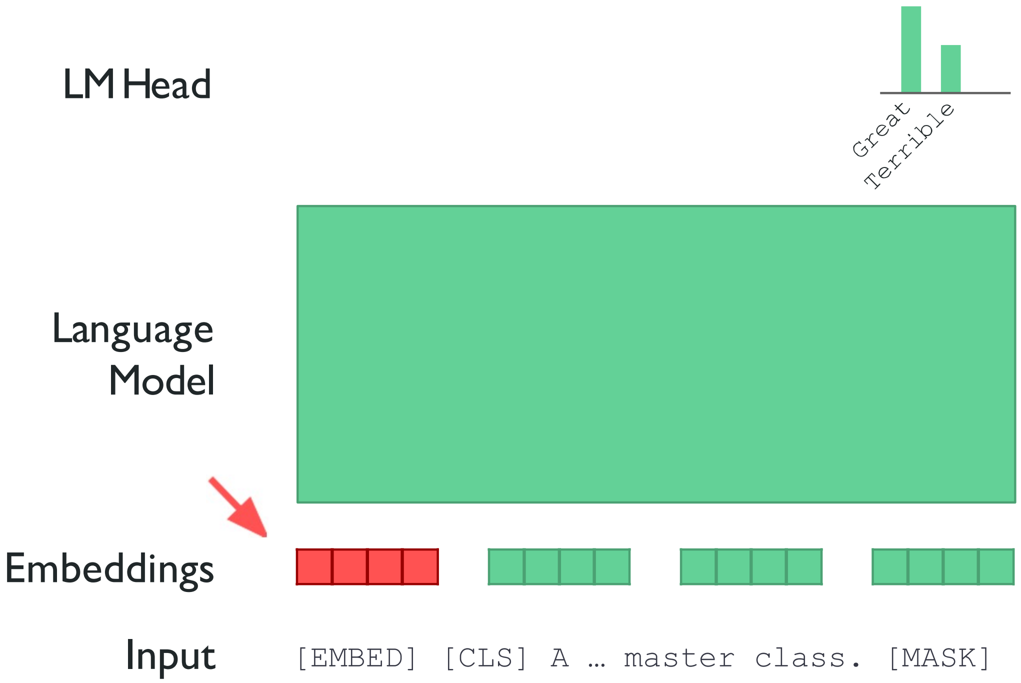
1. **Prompt search** methods try to learn the tokens in the prompt.



Input-level modifications

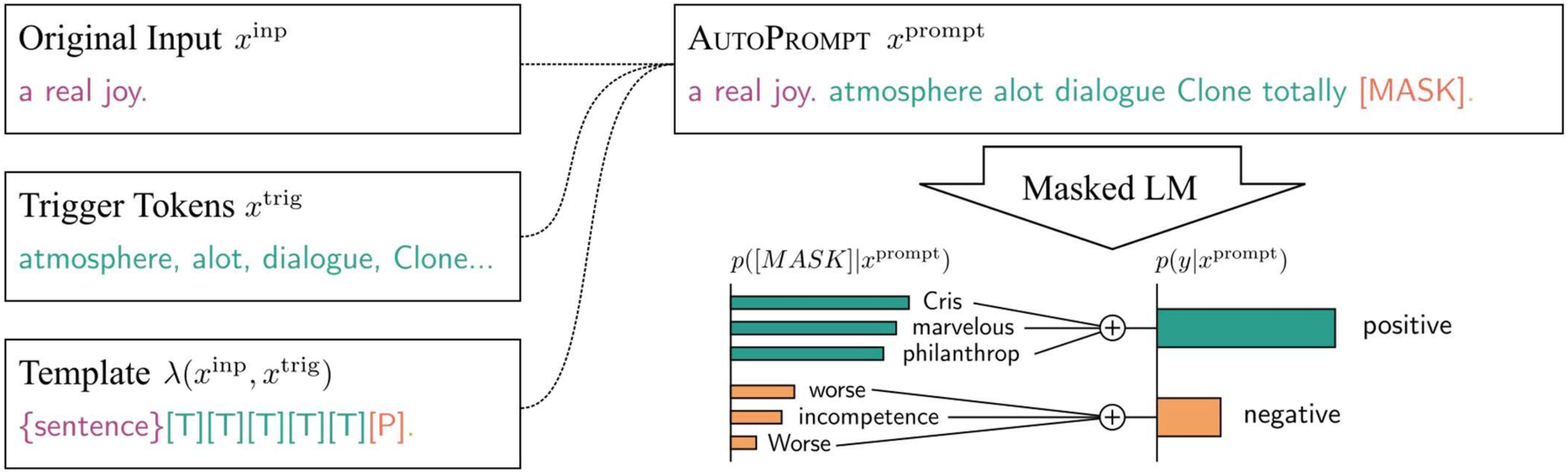
Two types:

1. Prompt search methods try to learn the tokens in the prompt.
2. **Prompt tuning** methods introduce novel embeddings that are learned using gradient descent.



Prompt Search Methods

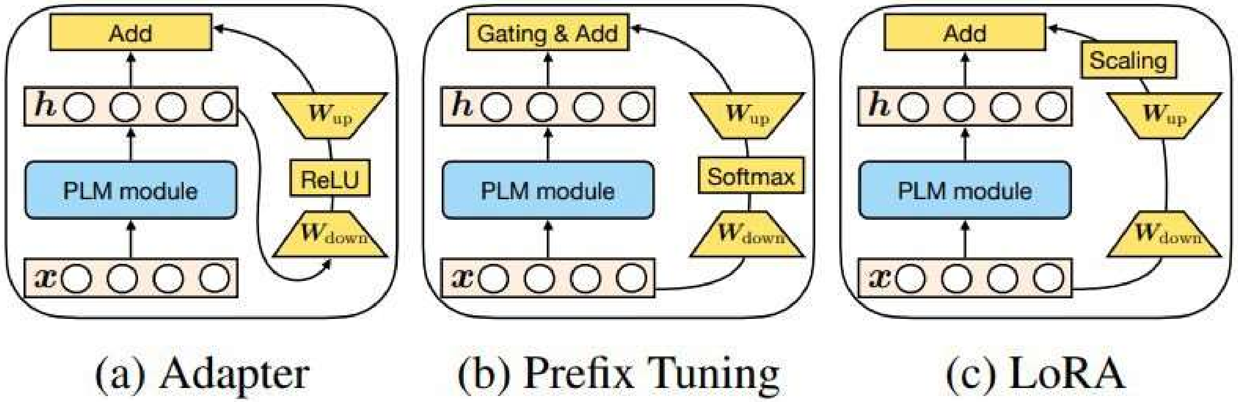
AutoPrompt: Iteratively updates tokens in the pattern using a gradient-guided search. (Shin et al. 2020)



Shin et al. 2020. "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts"

Towards a Unified View of Parameter-Efficient Transfer Learning

Approaches are more similar than at first glance.



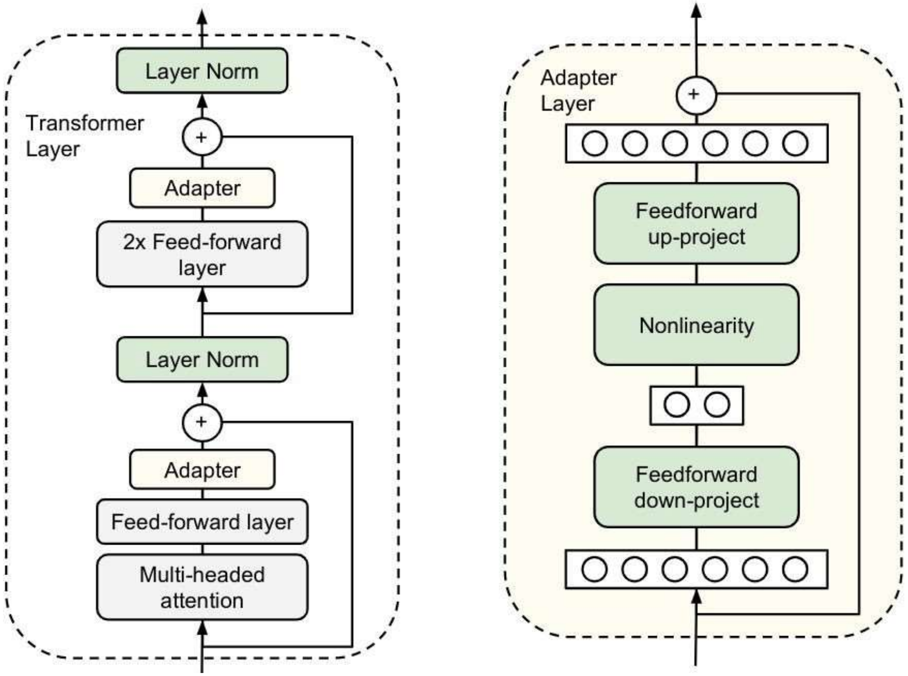
He et al., 2022. "Towards a Unified View of Parameter-Efficient Transfer Learning"

Adapters

Add MLP + skip connection after each feedforward layer.

MLP projects to a low dimensional space to reduce parameters.

Tune only the MLP layers on new tasks.



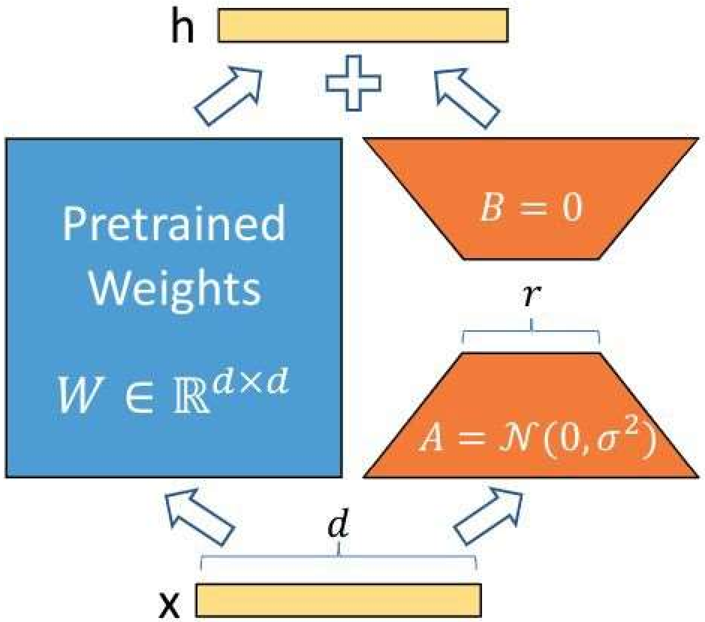
Houlsby et al., 2019. “Parameter-Efficient Transfer Learning for NLP”

LoRa

Low rank additive updates to model weights.

$W = W_0 + AB$

Where the rank of A and B << min(d,h)



Hu et al., 2021. "LoRA: Low-Rank Adaptation of Large Language Models"

ZEroShot learning from Task descriptions

Instead of instructions, use QA

100 information seeking *tasks*

Train it on some, test on others

"What camp zones are in this national park?"
"Does this national park have stores that sell firewood?"
"Does this national park have a gift shop selling handmade items?" "Wh
ere are bird watching spots near a lake in this national park?" "What ar
e the popular activities to do in the rivers at this national park?" "Is spelu
nking at this national park allowed?"
"Can you boat and grill at this national park?"
"How many people can fit in group campsites in this national park?"

"How long is the cave in this national park?"
"Could you mention the camp zones in this national park?"
"How many plants living inside this national park are endangered?"

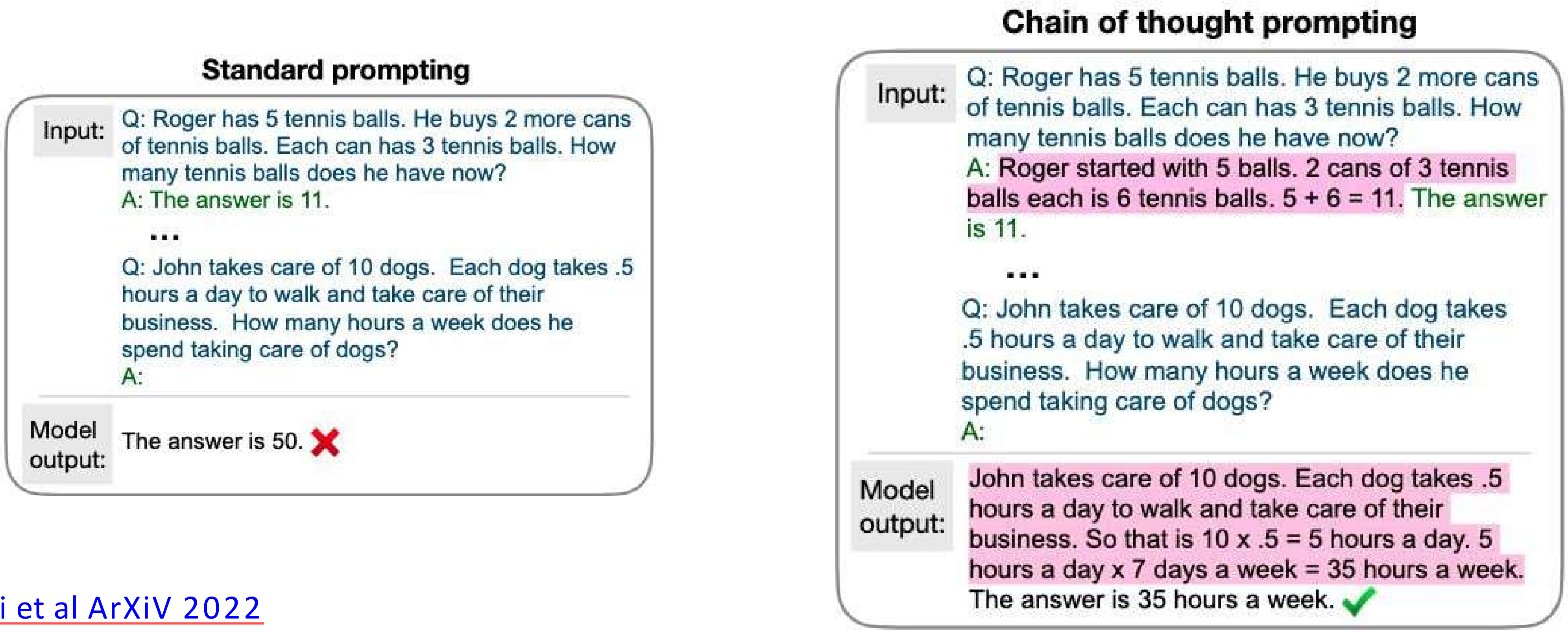
No examples at test-time
(Zero-shot!)

ZEST

Weller et al EMNLP 2020 (note, there are 2 ZESTs at that conference)

Chain of Thought Prompting

Give more “instructions” specific to the instance, only in-context learning



Wei et al ArXiV 2022

Automatic Chain of Thought

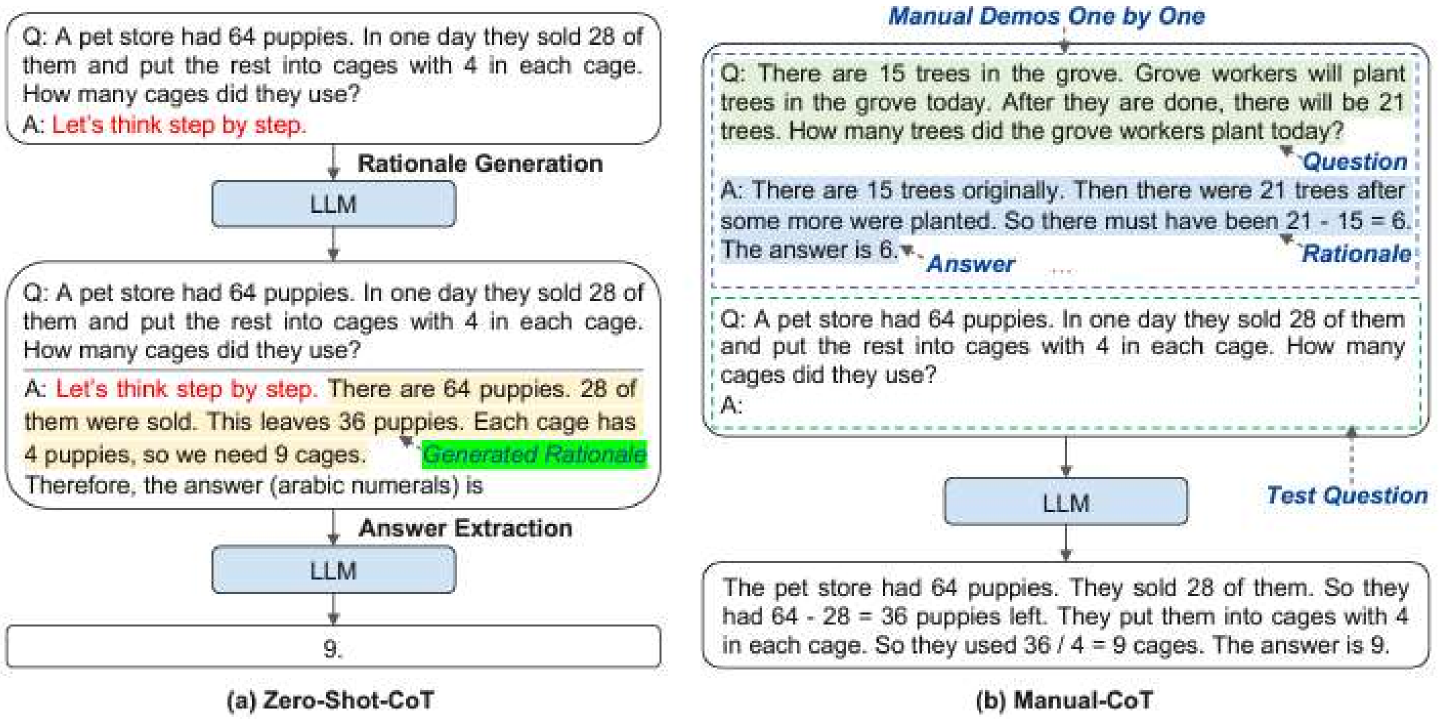


Figure 1: Zero-Shot-CoT [Kojima et al., 2022] (using the "Let's think step by step" prompt) and Manual-CoT [Wei et al., 2022a] (using manually designed demonstrations one by one) with example inputs and outputs of an LLM.