Secrets in Training a Large Language Model

Houston Machine Learning Meetup
Online Event

July 29, 2PM CDT, 2023

Yan Xu
Youtube: YanAITalk
https://medium.com/@YanAIx

History of LLM

Attention

Transformer

BERT
340 mil

GPT-1
117 mil

GPT-2
1.5B

GPT-3
175B

LaMDA (Bard) - 173B
MT-NLG (Megatron-Turing) – 530B

ChatGPT

PaLM – 540B

BLOOM – 176B
Flan – 20B
YaLM – 100B

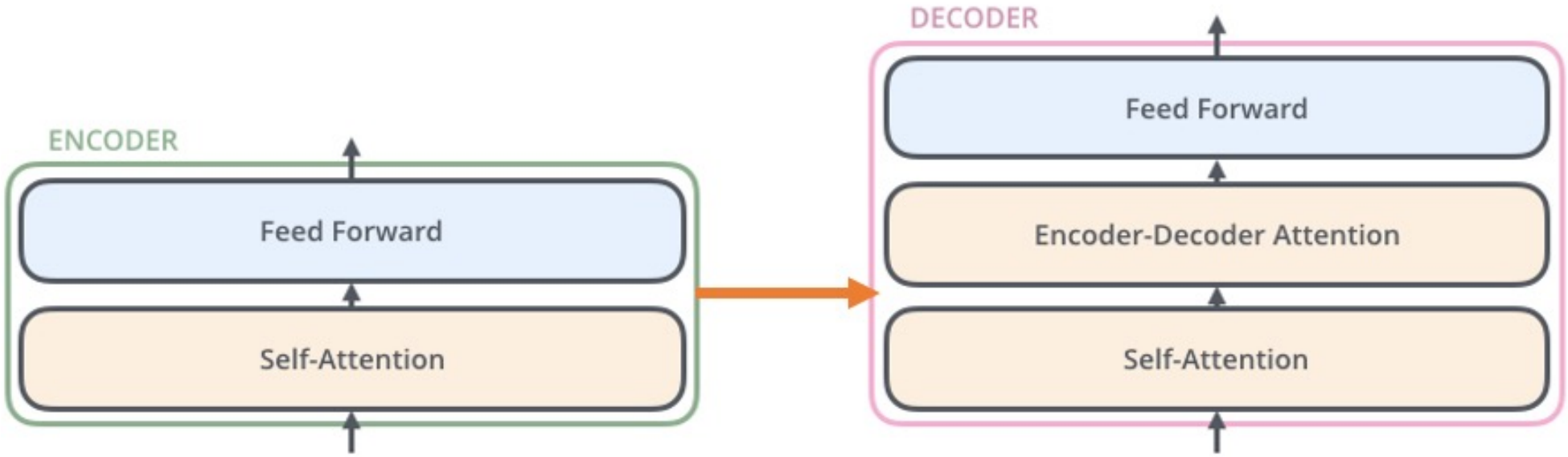
GPT-4
MiniGPT-4
LLaMA 1 (7-65B, non-commercial)
- Alpaca (7B)
- Vicuna (13B)
- OpenLLaMa (7B)

LLaMA 2 (7-70B)
PaLM 2 (up to 340B)

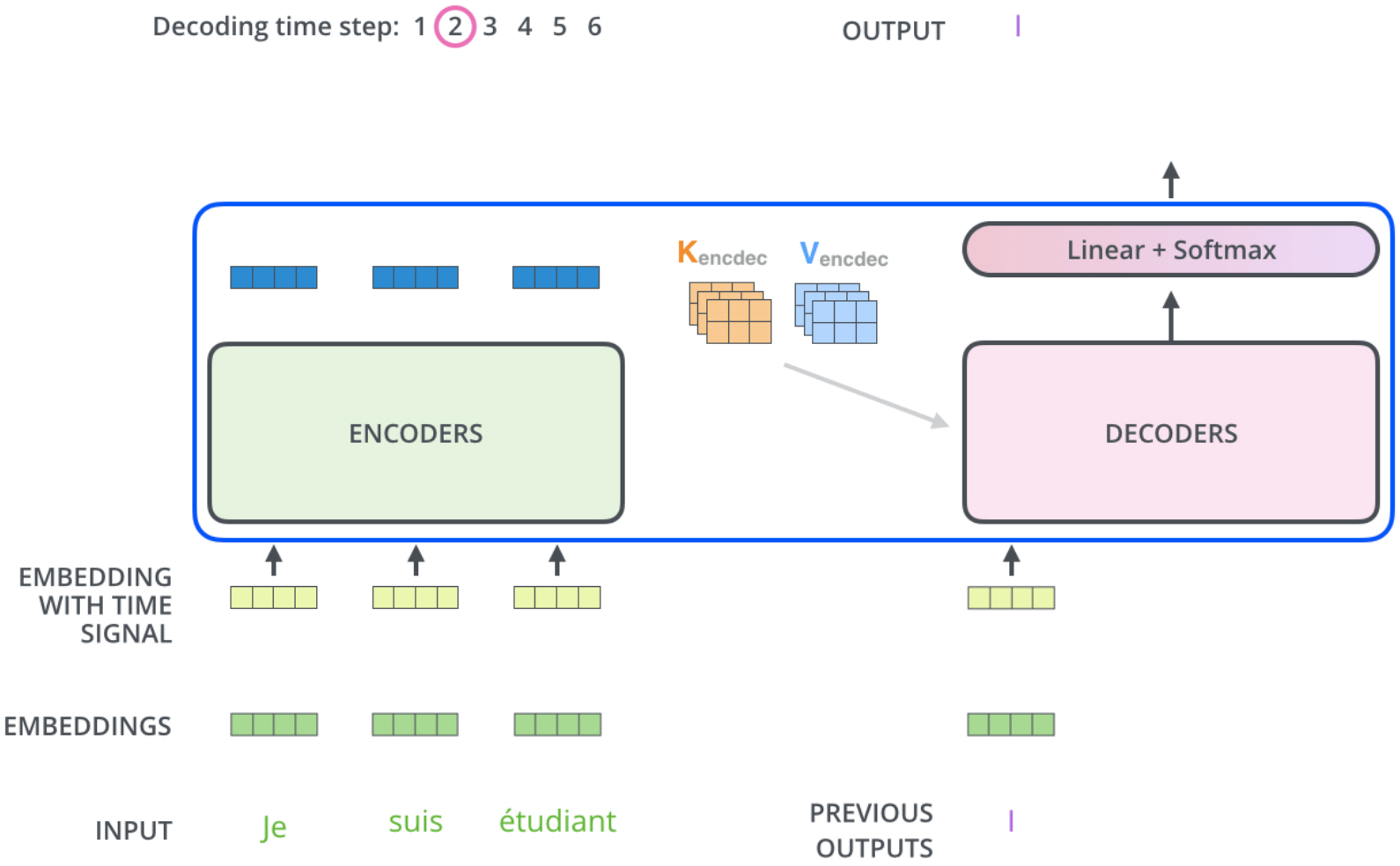
Research purpose only
Research and commercial use

Transformer

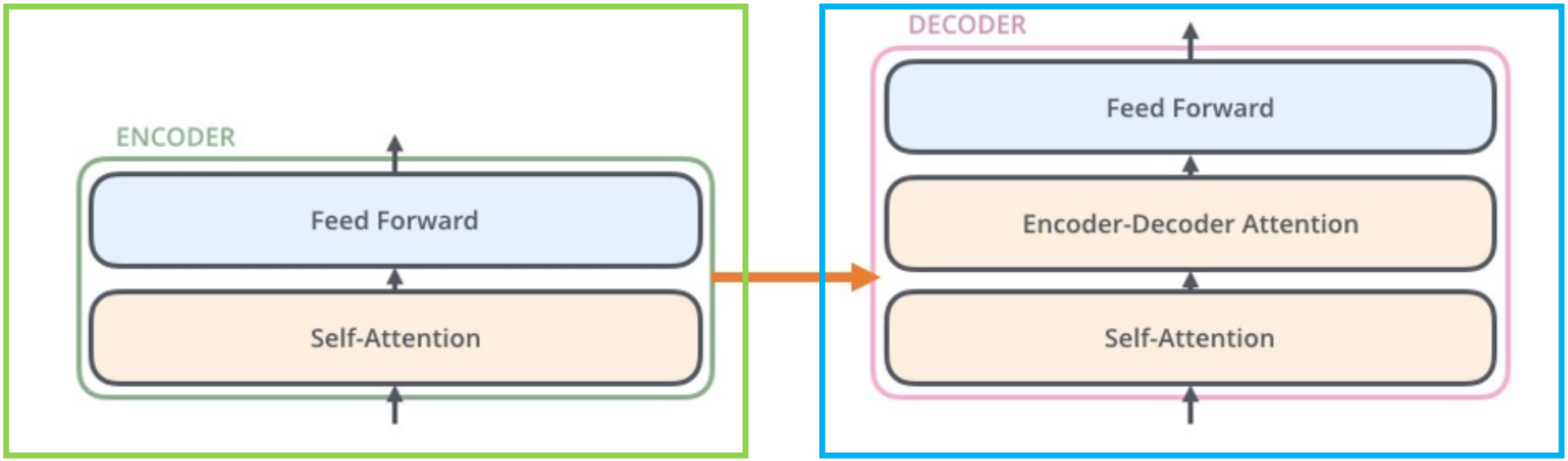
https://medium.com/@YanAIx/step-by-step-into-transformer-79531eb2bb84



Transformer



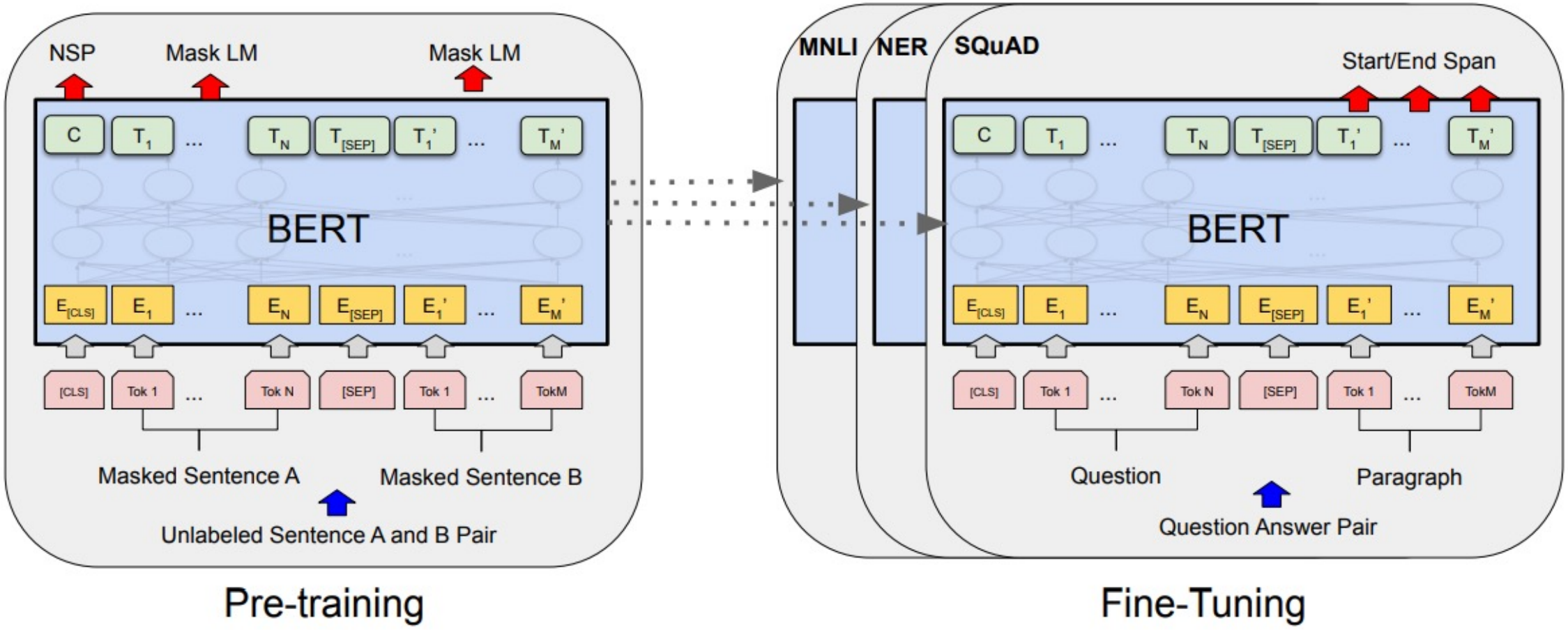
Transformer



BERT: Bidirectional Encoder Representations from Transformers

GPT: Generative Pretrained Transformer

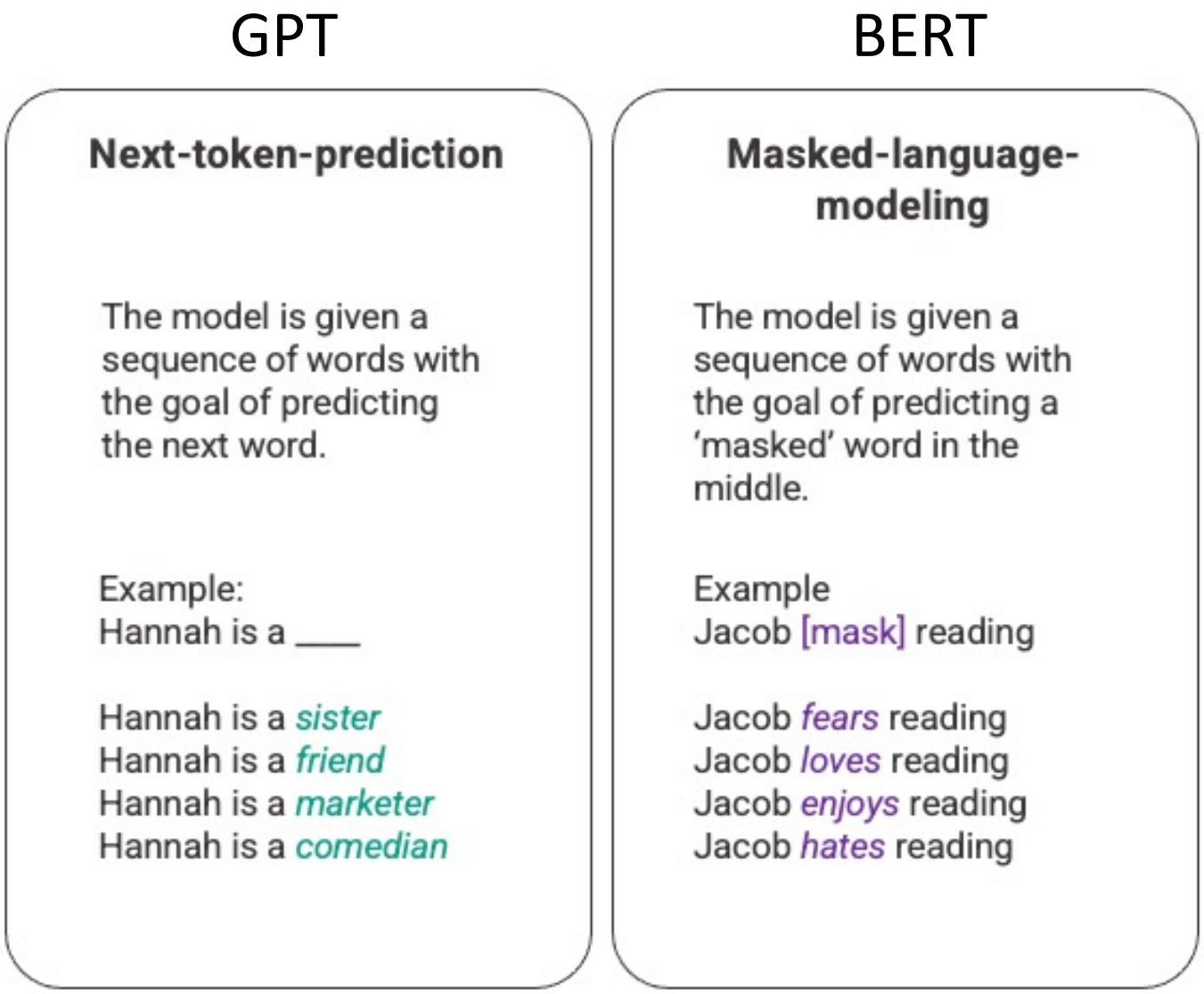
Pretraining and Fine-tuning



Language understanding

Adapting to different tasks

Pre-training

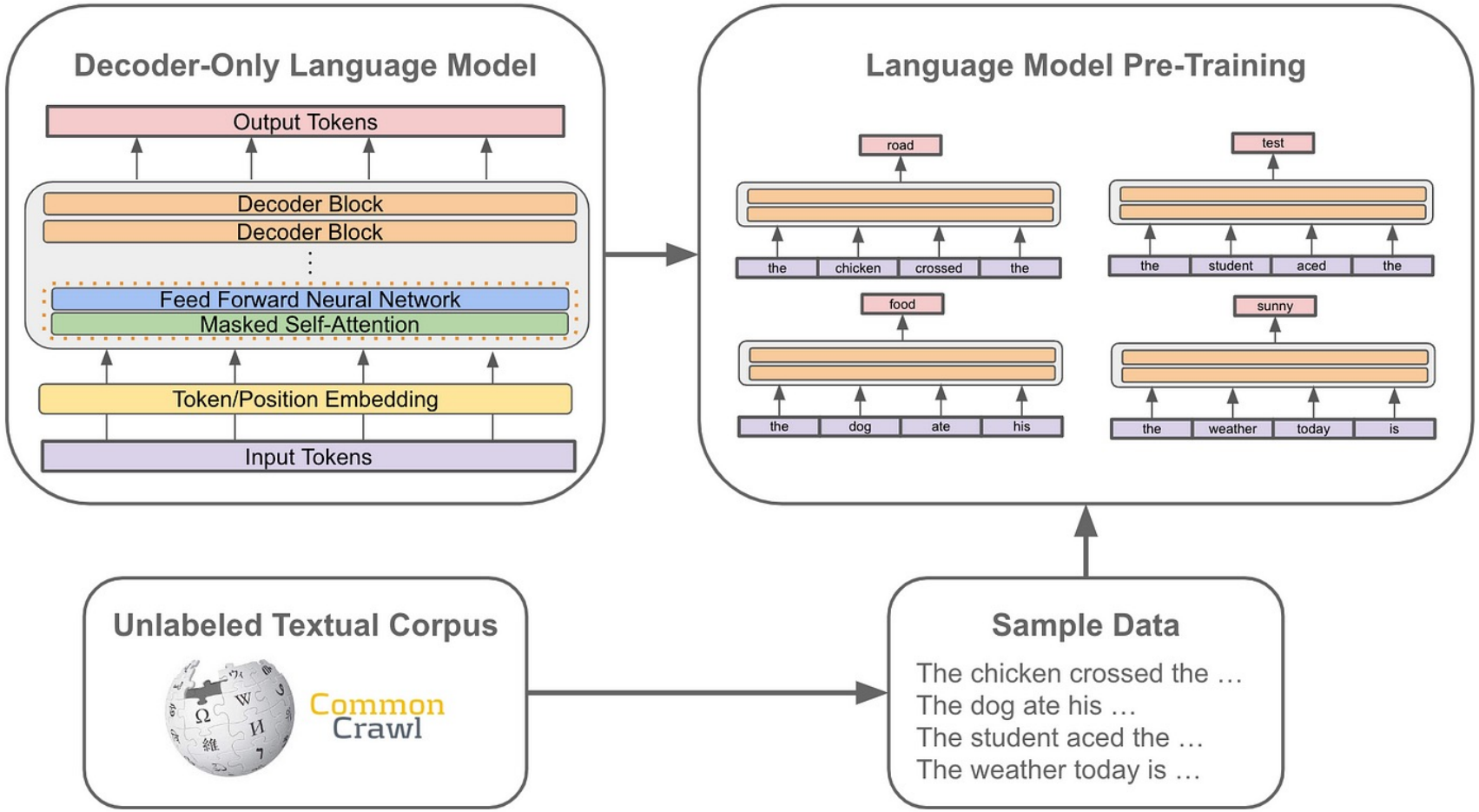


GPT Pretraining

$$\mathcal{L}(\mathcal{U}) = \sum_{i=1}^{N} \log\left(\mathbb{P}(u_i | u_{i-k}, \ldots, u_{i-1}, \Theta)\right)$$

Language model loss over the full text corpus

Conditional probability of i-th token given k preceding tokens and model parameters θ

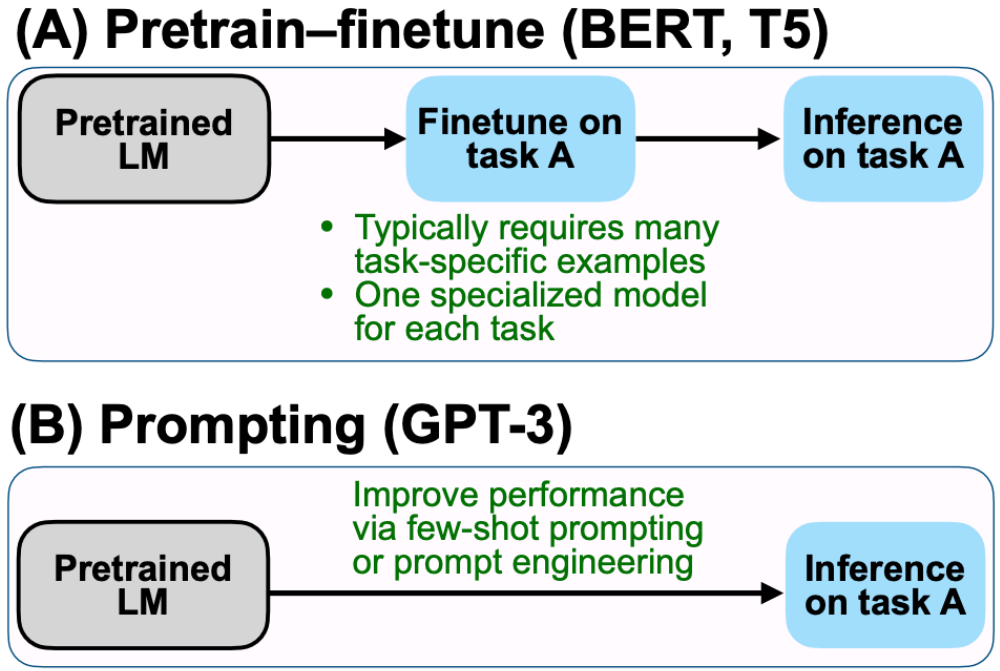


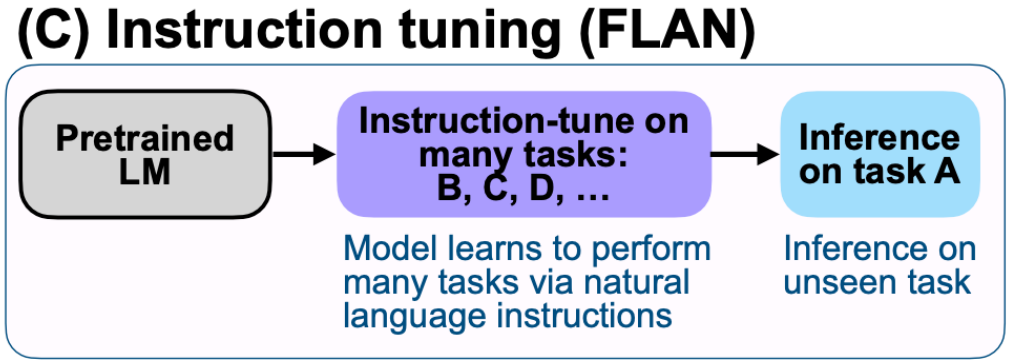
Pretraining

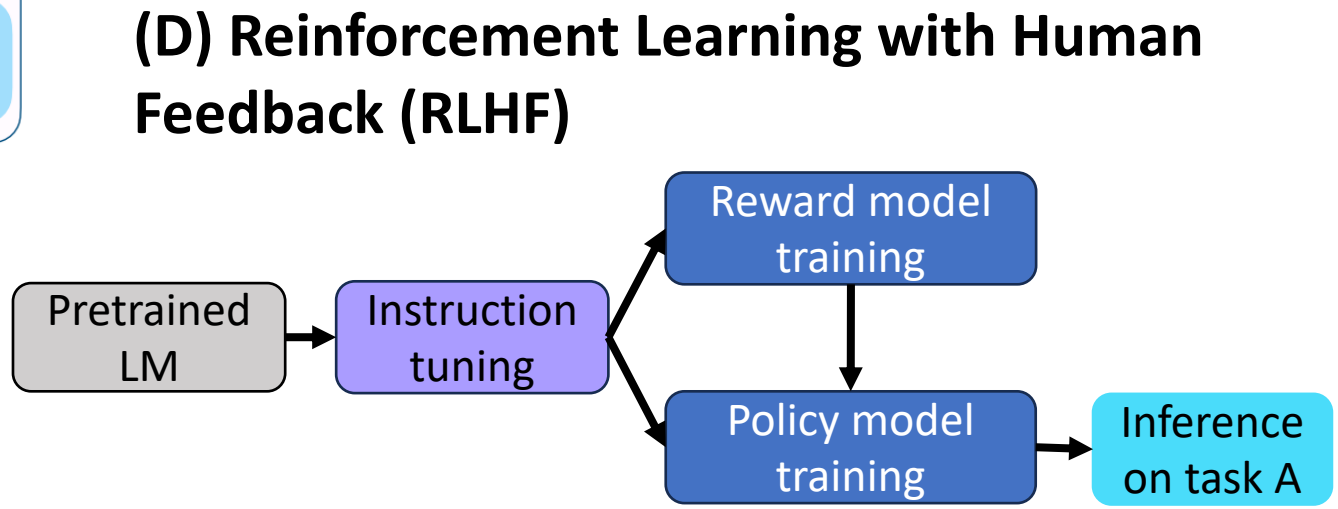
Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table 2.2: Datasets used to train GPT-3. "Weight in training mix" refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

Fine-tuning


(A) Pretrain–finetune (BERT, T5)
Pretrained LM
Finetune on task A
Inference on task A
Typically requires many task-specific examples
One specialized model for each task
(B) Prompting (GPT-3)
Pretrained LM
Improve performance via few-shot prompting or prompt engineering
Inference on task A


(C) Instruction tuning (FLAN)
Pretrained LM
Instruction-tune on many tasks: B, C, D, …
Inference on task A
Model learns to perform many tasks via natural language instructions
Inference on unseen task


(D) Reinforcement Learning with Human Feedback (RLHF)
Pretrained LM
Instruction tuning
Reward model training
Policy model training
Inference on task A

Prompting

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

The three settings we explore for in-context learning

Zero-shot

Prompting

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:        ← task description
cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← example
cheese =>                           ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                           ← prompt
```

Instruction tuning

Finetune on many tasks ("instruction-tuning")

Input (Commonsense Reasoning)

Here is a goal: Get a cool sleep on summer days.

How would you accomplish this goal?

OPTIONS:

-Keep stack of pillow cases in fridge.

-Keep stack of pillow cases in oven.

Target

keep stack of pillow cases in fridge

Input (Translation)

Translate this sentence to Spanish:

The new office building was built in less than three months.

Target

El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks

Coreference resolution tasks

...

Inference on unseen task type

Input (Natural Language Inference)

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

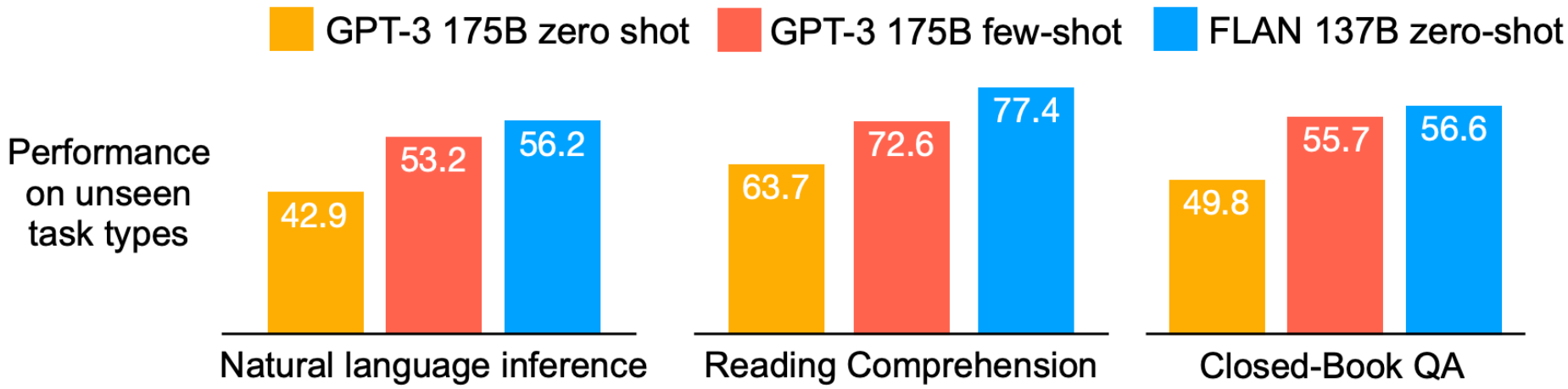
Does the premise entail the hypothesis?

OPTIONS:

-yes -it is not possible to tell -no

FLAN Response

It is not possible to tell



Instruction fine-tuning

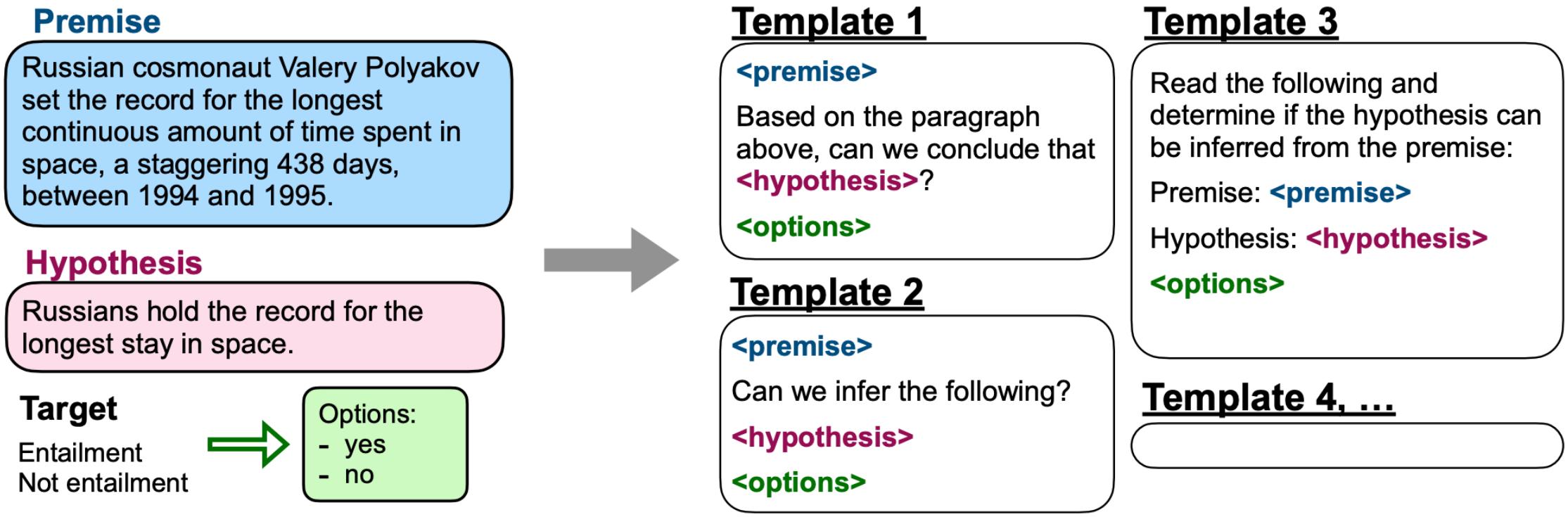
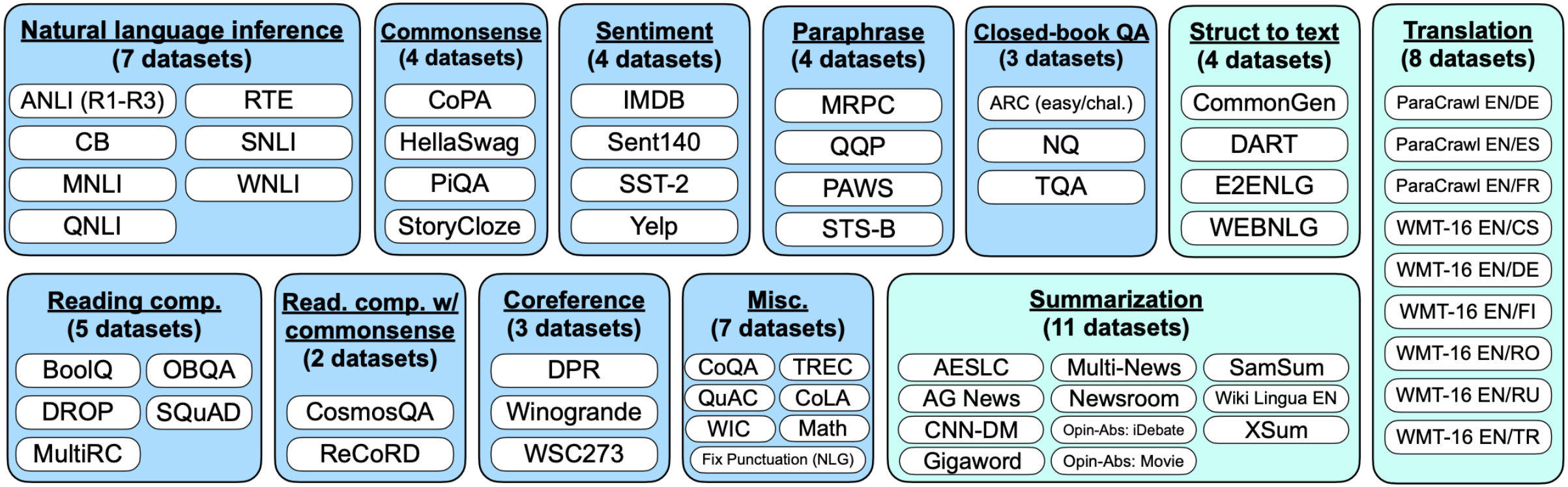


Figure 4: Multiple instruction templates describing a natural language inference task.

Datasets and task clusters



Instruction fine-tuning

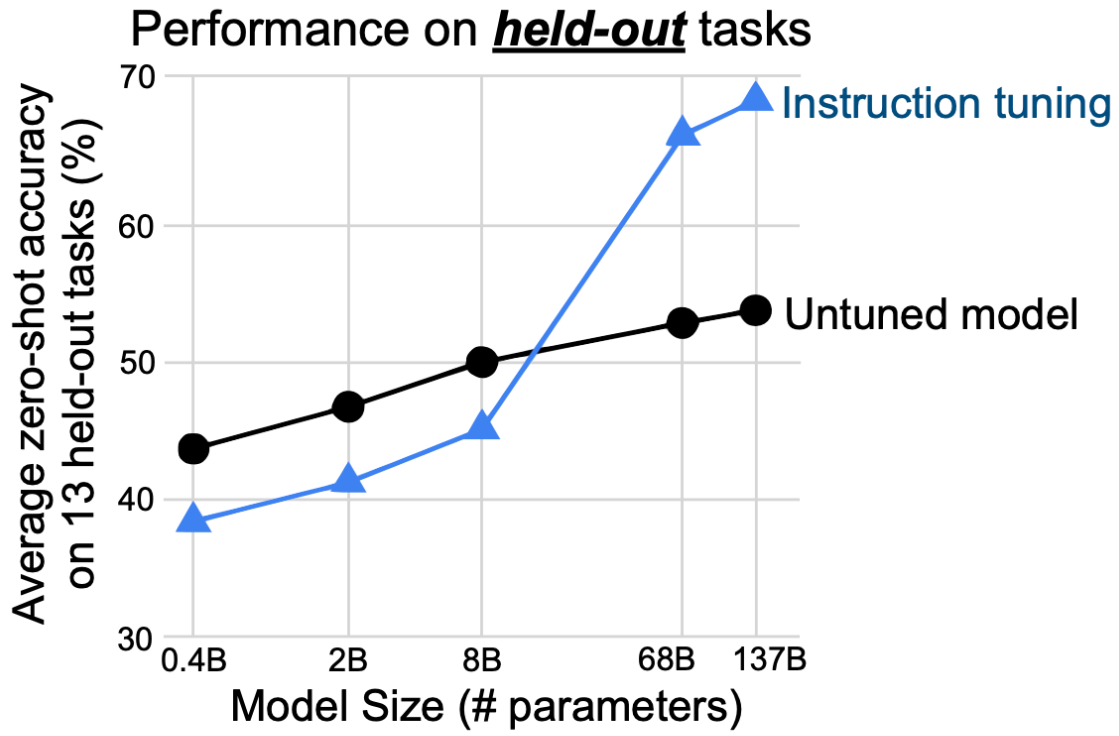
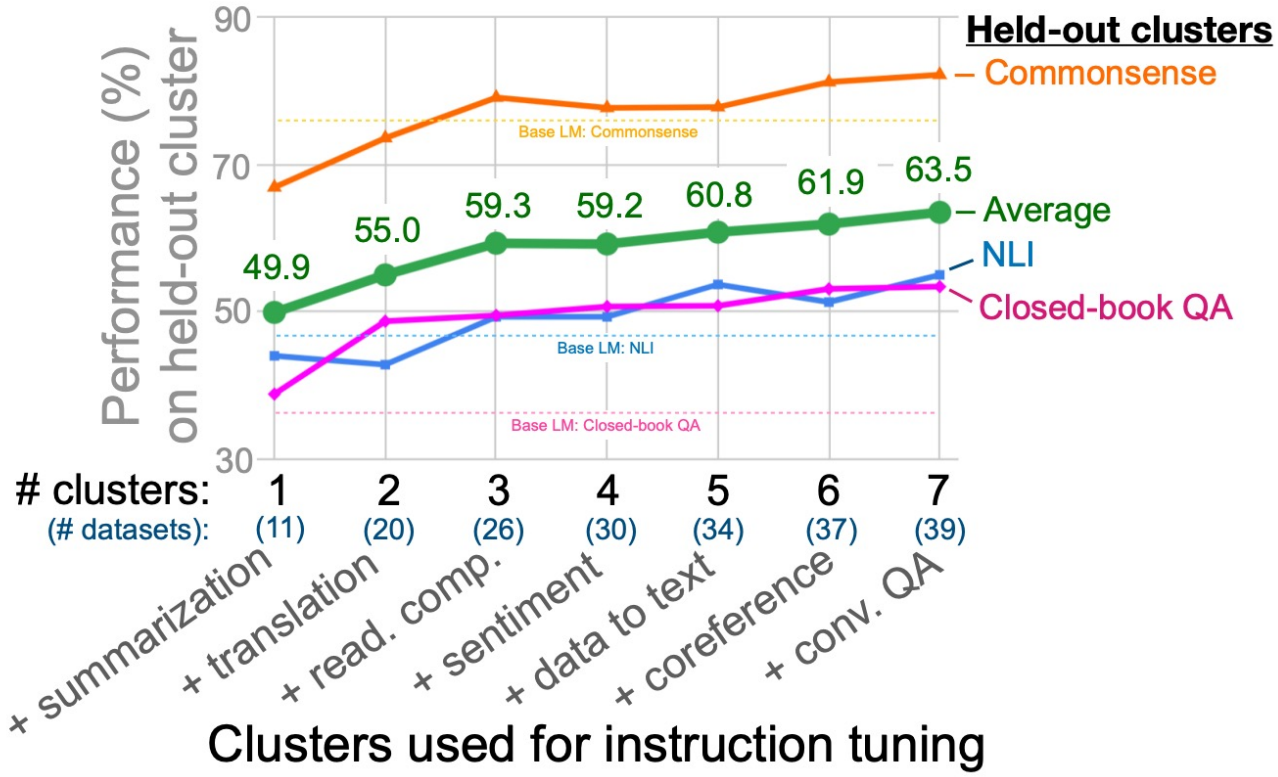


Figure 7: Whereas instruction tuning helps large models generalize to new tasks, for small models it actually hurts generalization to unseen tasks, potentially because all model capacity is used to learn the mixture of instruction tuning tasks.



Efficient fine-tuning of LLM

- LoRA: Low-Rank Adaption of LLM

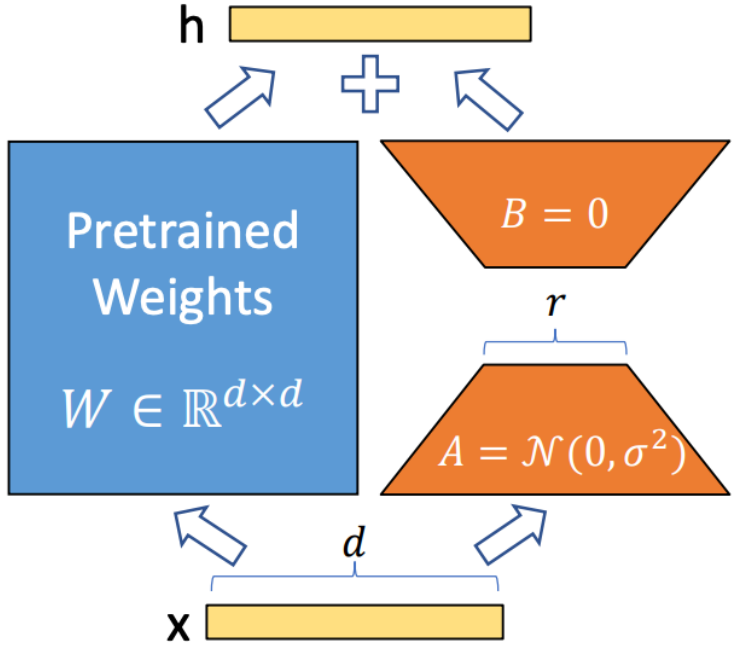


Figure 1: Our reparametrization. We only train A and B.

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

Table 4: Performance of different adaptation methods on GPT-3 175B. We report the logical form validation accuracy on WikiSQL, validation accuracy on MultiNLI-matched, and Rouge-1/2/L on SAMSum. LoRA performs better than prior approaches, including full fine-tuning. The results on WikiSQL have a fluctuation around $\pm 0.5\%$, MNLI-m around $\pm 0.1\%$, and SAMSum around $\pm 0.2/\pm 0.2/\pm 0.1$ for the three metrics.

Instruction finetuning is highly effective but it has inherent limitations

What is the learning objective in instruction finetuning?

For a given input, the target is the single correct answer

In Reinforcement Learning, this is called “behavior cloning”

Hope is that if we have enough of these, the model can learn to generalize

This requires formalizing the correct behavior for a given input

Observations

Increasingly we want to teach models more abstract behaviors

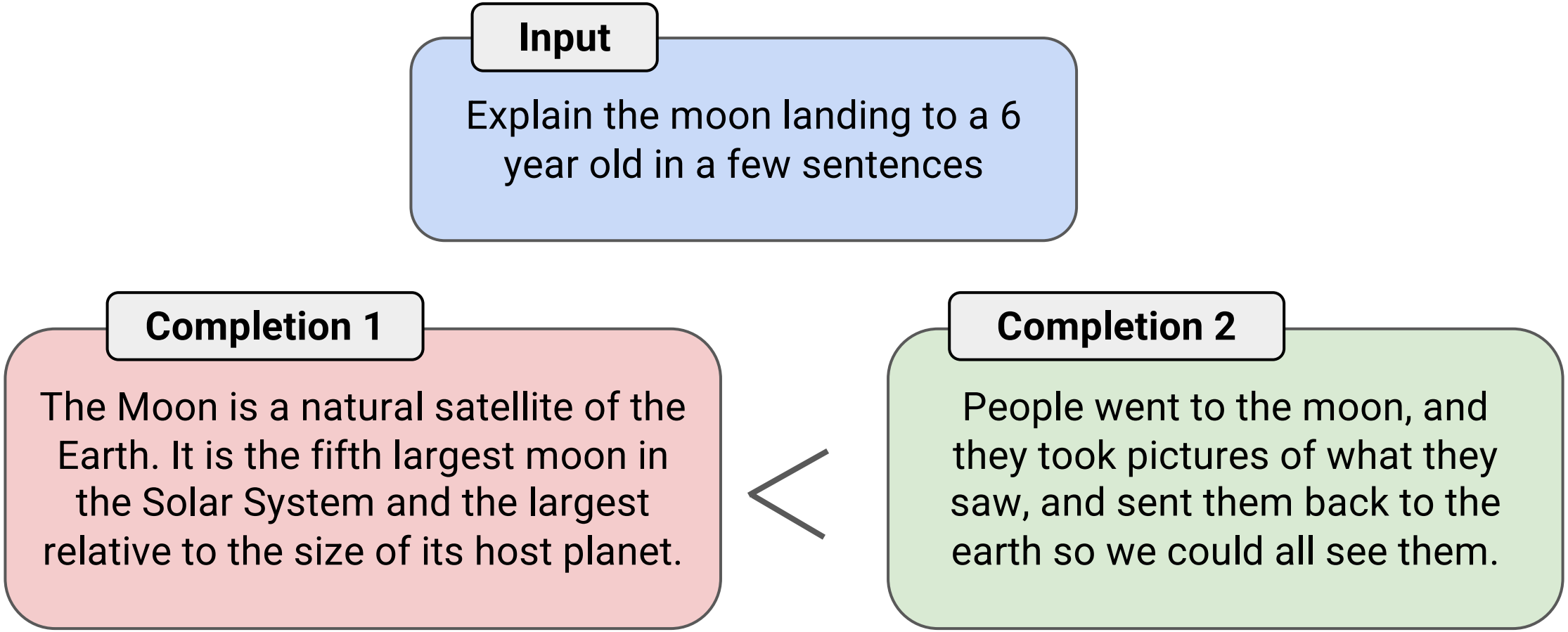
Objective function of instruction finetuning seems to be the “bottleneck” of teaching these behaviors

The maximum likelihood objective is “predefined” function (i.e. no learnable parameter)

Can we parameterize the objective function and ***learn*** it?

-> **Reinforcement Learning with Human Feedback (RLHF)**

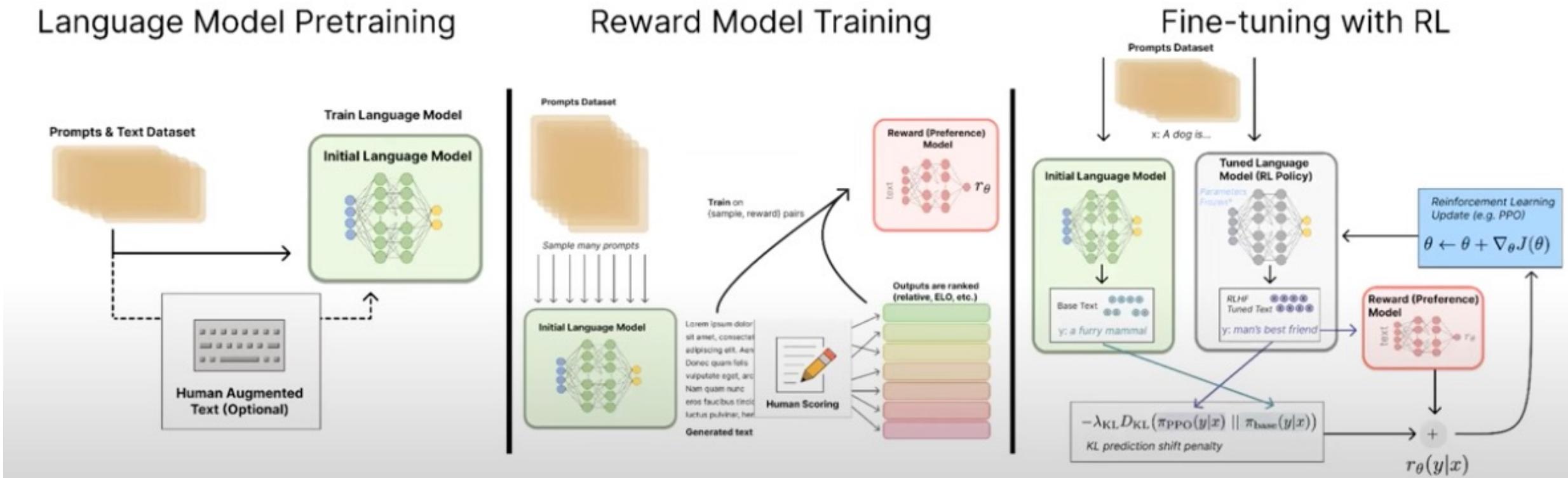
Reward Model (RM) training data: which completion is better?



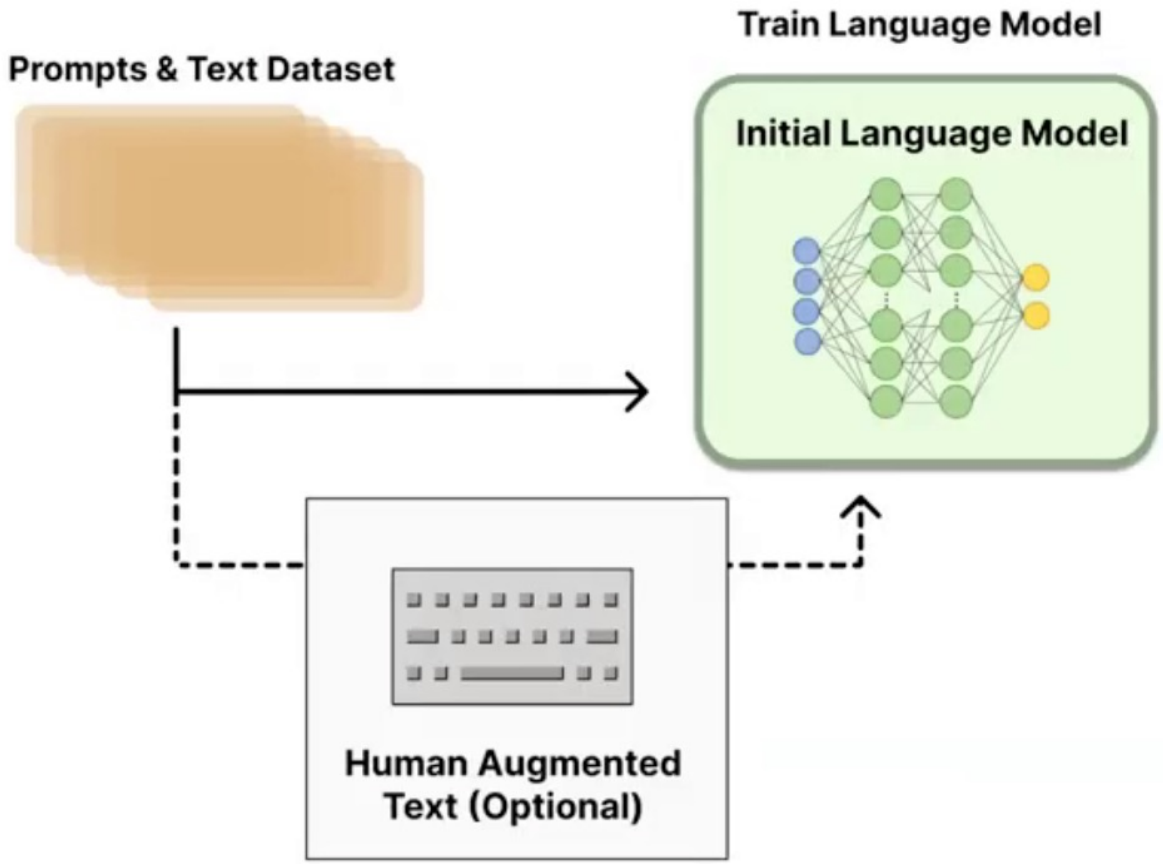
Humans label which completion is preferred.

This setup aims to align models to the human preference

Modern RLHF overview

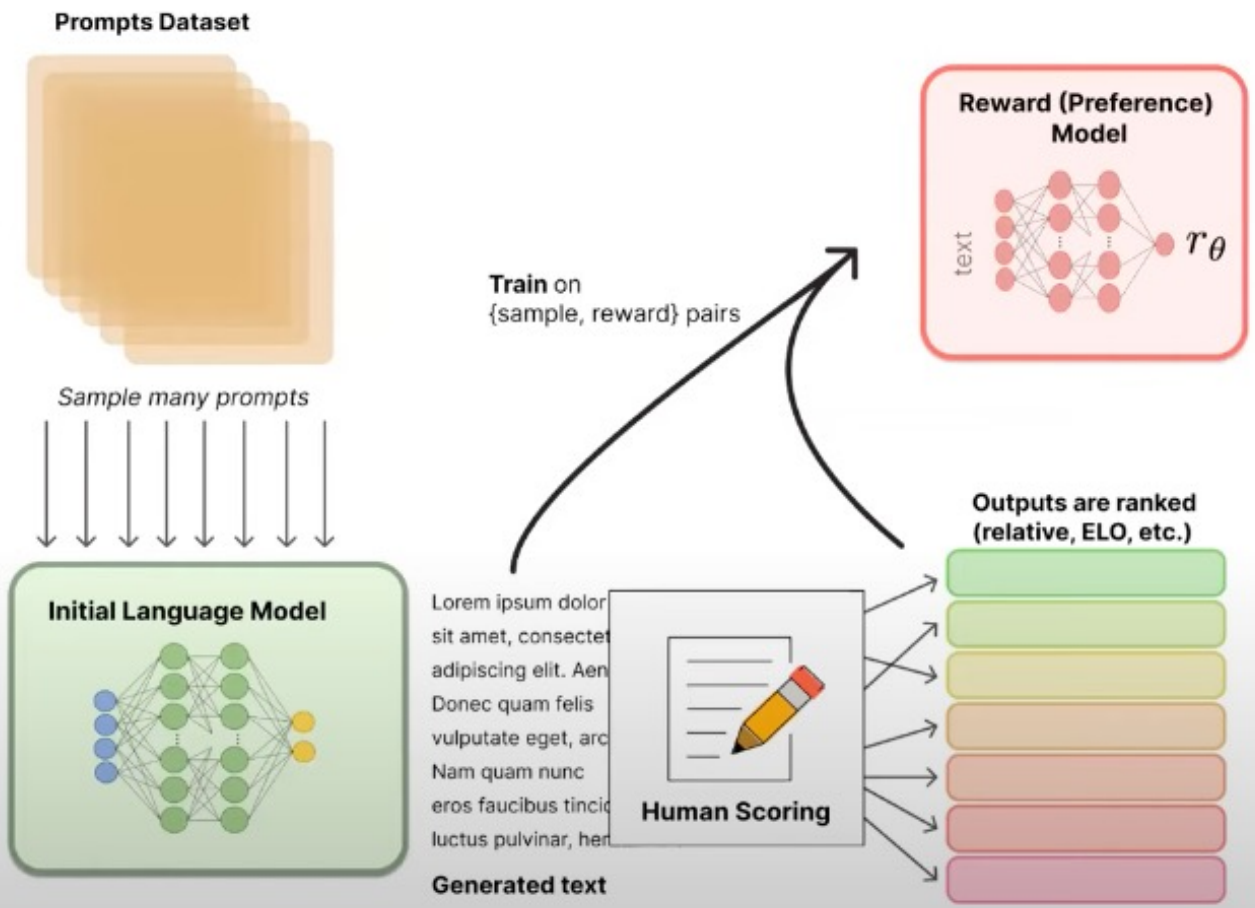


Language Model Pretraining



- Uses common training techniques in NLP
- Base model strength determines need for supervised fine tuning / annotations
- Open source models ~1.5yr behind closed models (for now!)

Reward Model (RM) training



How to capture human sentiments in samples and curated text? What is the loss!

Goal: get a model that maps

input text → scalar reward

Reward Model (RM) training objective function

Let p_{ij} be the probability that completion y_i is better than completion y_j

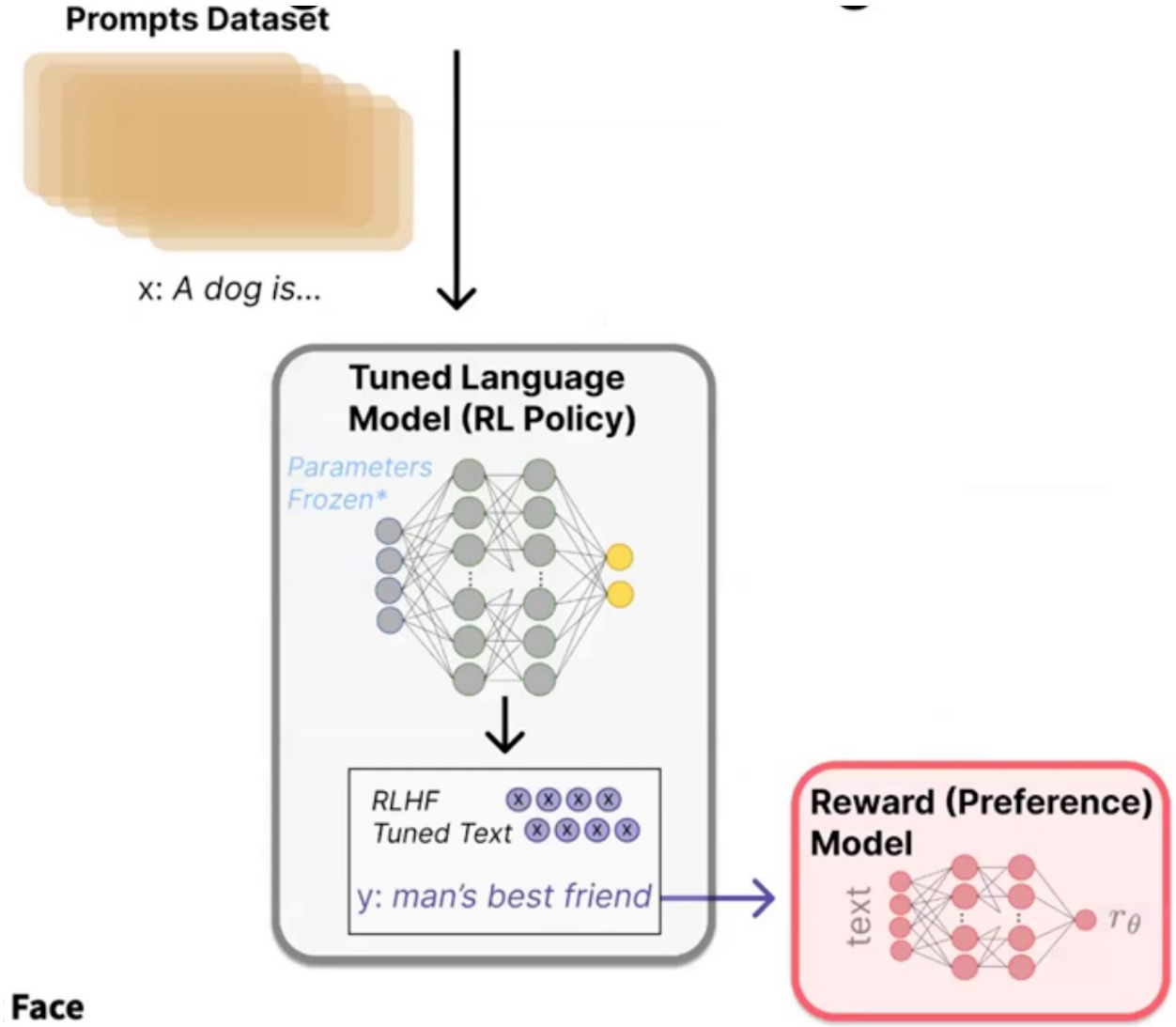
Bradley–Terry model (1952): log odds that completion y_i is favored over y_j is modeled as difference in the rewards

$$\log \frac{p_{ij}}{1 - p_{ij}} = r(x, y_i; \phi) - r(x, y_j; \phi)$$

$$p_{ij} = \frac{e^{r(x,y_i;\phi)-r(x,y_j;\phi)}}{1 + e^{r(x,y_i;\phi)-r(x,y_j;\phi)}} = \sigma(r(x, y_i; \phi) - r(x, y_j; \phi))$$

$$\max_{\phi} \sum_{x,y_i,y_j \in D} \log p_{ij}$$

Policy Model Training



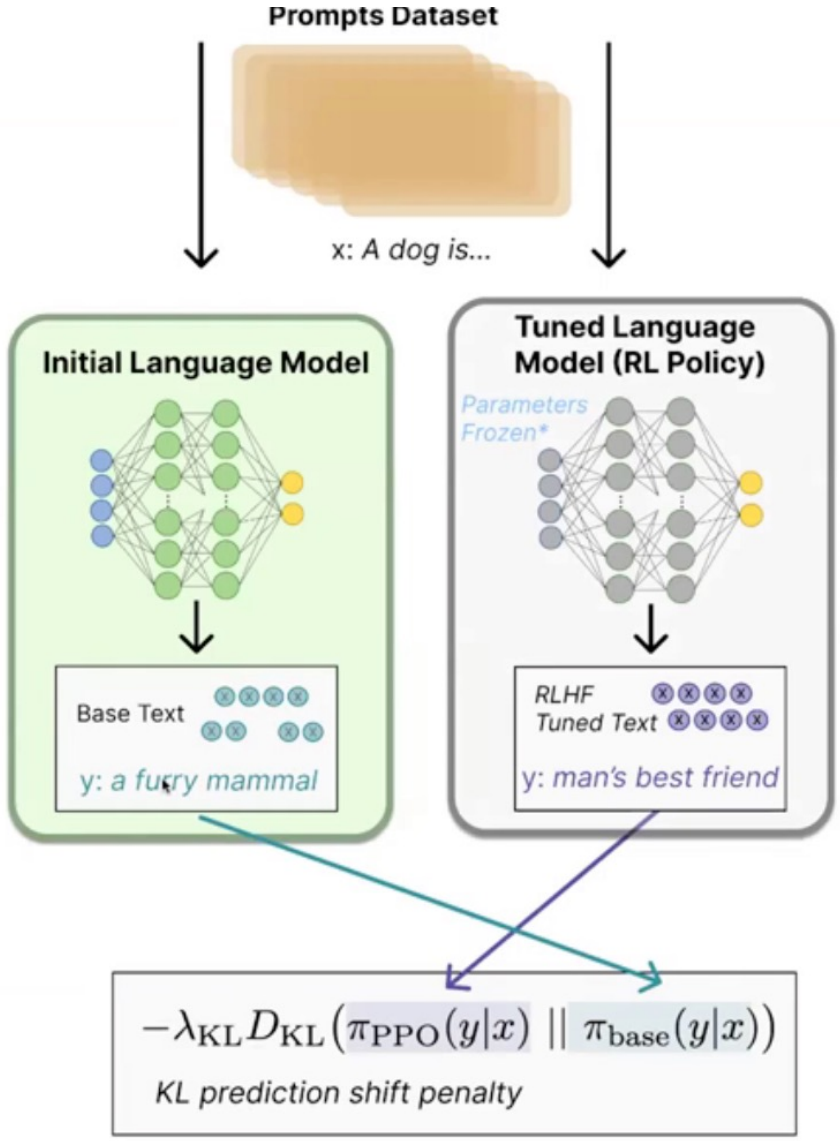
Face

Policy Model Training

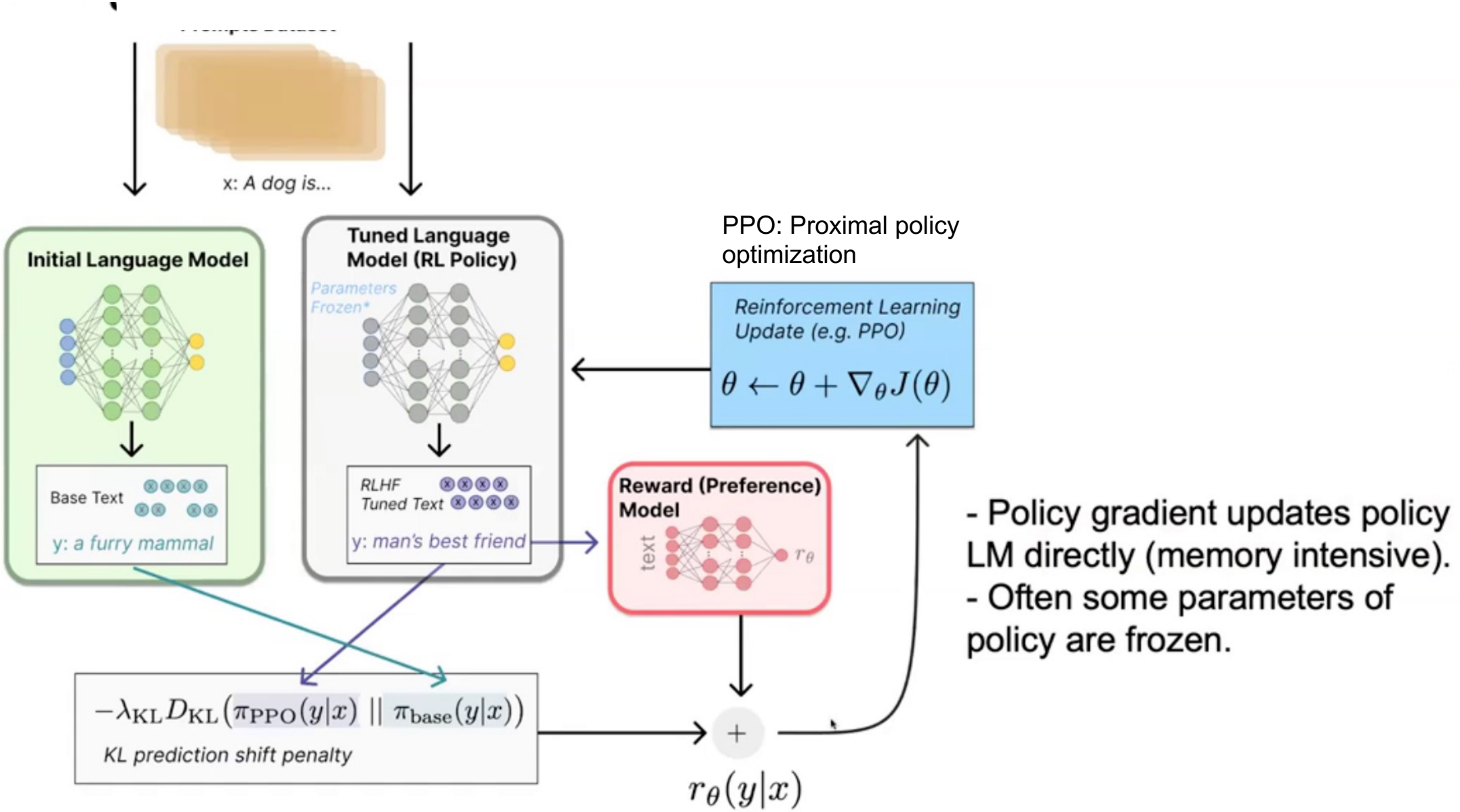
Kullback–Leibler (KL) divergence: $D_{\mathrm{KL}}(P \parallel Q)$
Distance between distributions

Constrains the RL fine-tuning to not result in a LM that outputs gibberish (to fool the reward model).

Note: DeepMind did this in RL Loss (not reward), see GopherCite



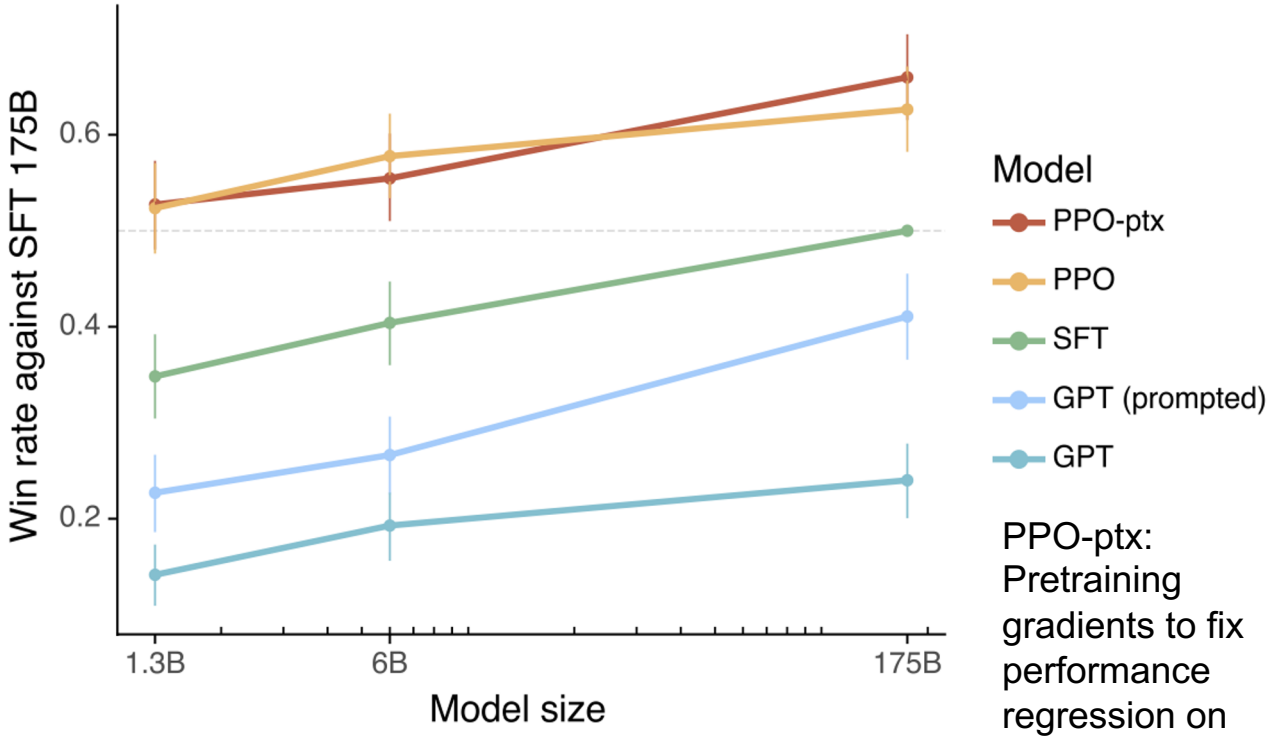
Policy Model Training



PPO: Proximal policy optimization

- Policy gradient updates policy LM directly (memory intensive).
- Often some parameters of policy are frozen.

RLHF models are *preferred* over instruction finetuned models



PPO-ptx: Pretraining gradients to fix performance regression on public

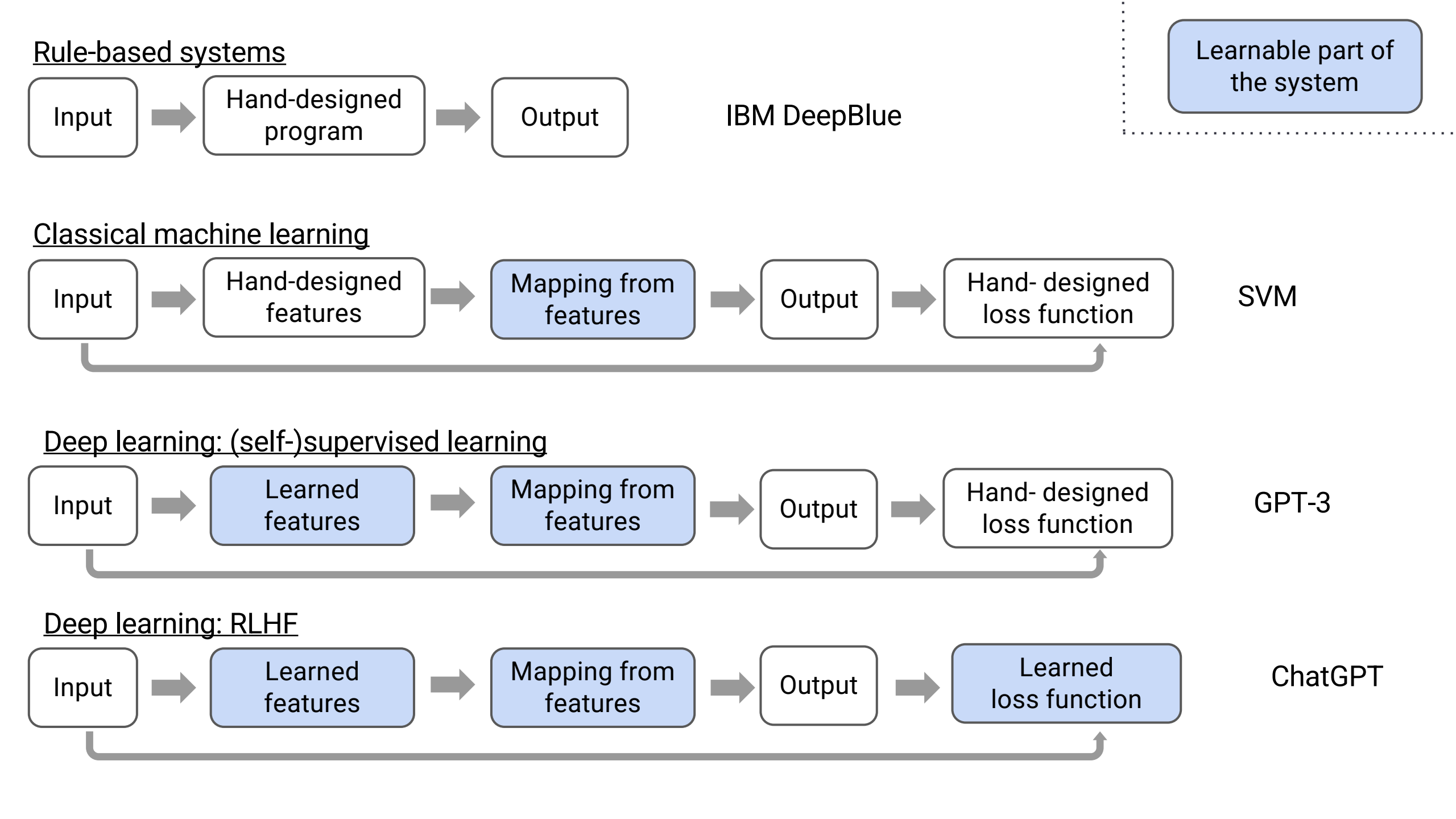
Prompt *Explain the moon landing to a 6 year old in a few sentences.*

Completion GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few
sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw,
and sent them back to the earth so we could all see them.
```


Learnable part of the system
Rule-based systems
Input
Hand-designed program
Output
IBM DeepBlue
Classical machine learning
Input
Hand-designed features
Mapping from features
Output
Hand- designed loss function
SVM
Deep learning: (self-)supervised learning
Input
Learned features
Mapping from features
Output
Hand- designed loss function
GPT-3
Deep learning: RLHF
Input
Learned features
Mapping from features
Output
Learned loss function
ChatGPT

Summary

Pretraining

Next-token-prediction

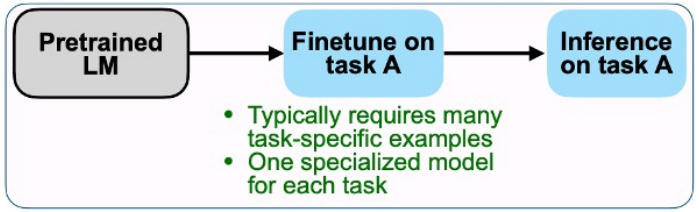
The model is given a sequence of words with the goal of predicting the next word.

Example:
Hannah is a ____

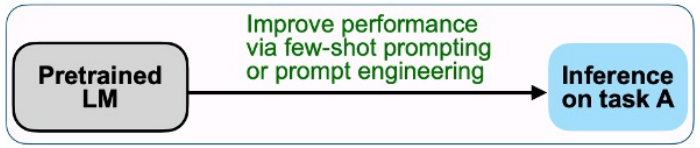
Hannah is a *sister*
Hannah is a *friend*
Hannah is a *marketer*
Hannah is a *comedian*

Fine-tuning

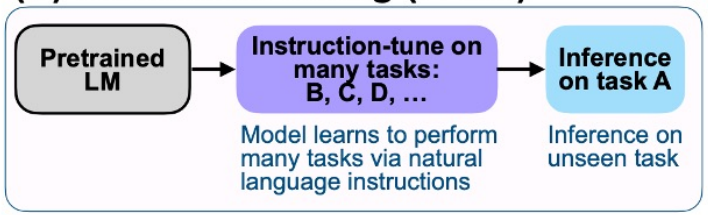
(A) Pretrain–finetune (BERT, T5)



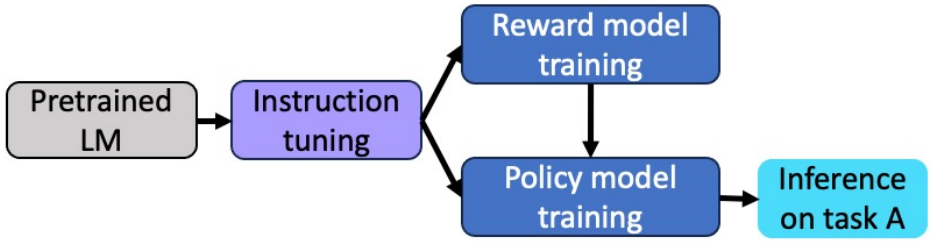
(B) Prompting (GPT-3)



(C) Instruction tuning (FLAN)



(D) Reinforcement Learning with Human Feedback (RLHF)



How to connect

- Meetup discussion and message: https://www.meetup.com/houston-machine-learning/
- Recordings will be posted at: https://www.youtube.com/@yanaitalk/videos
- Presentations posted at: https://medium.com/@YanAIx

Introduction to Vision Transformer (ViT)