Natural Language Understanding, Generation, and Machine Translation

Lecture 21: LLM Alignment and Evaluation

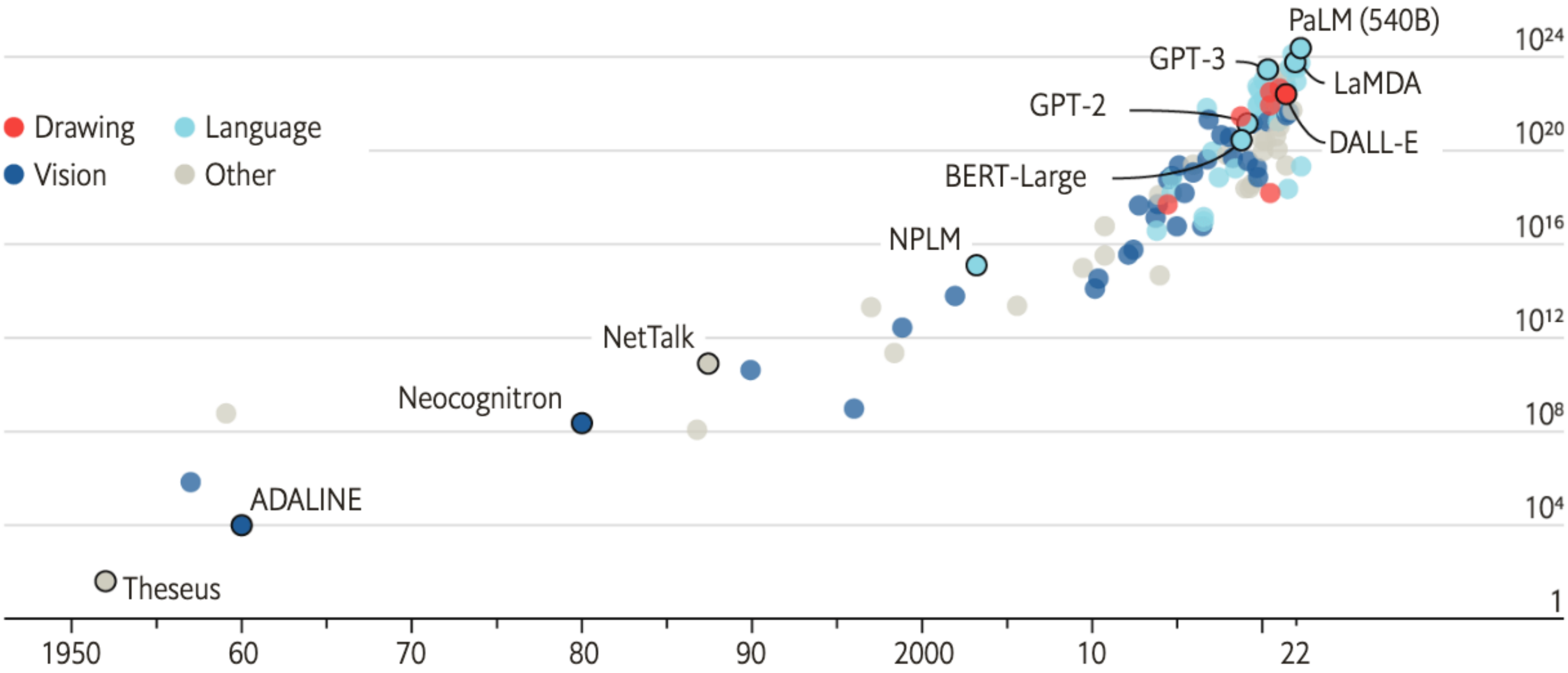
Pasquale Minervini
p.minervini@ed.ac.uk
March 8th, 2024

Large Language Models

The blessings of scale

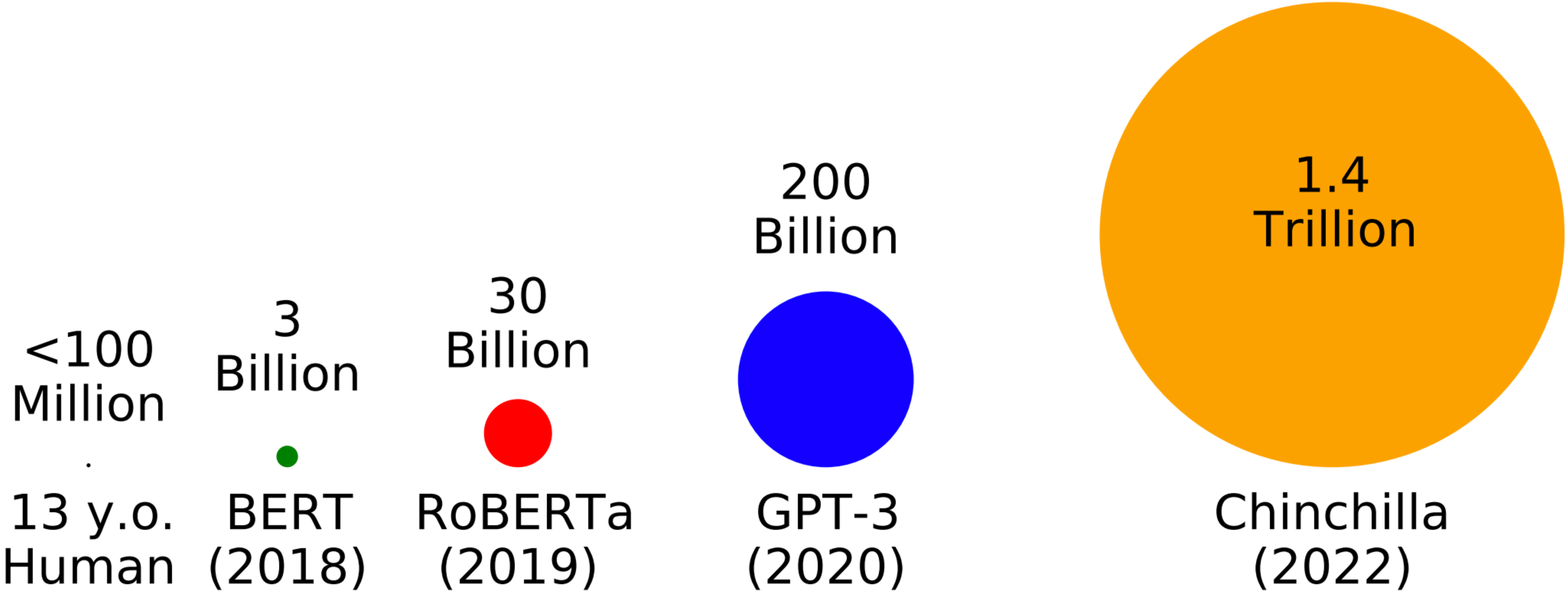
AI training runs, estimated computing resources used

Floating-point operations, selected systems, by type, log scale



Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data

Large Language Models



Number of tokens observed during “training”

Large Language Models

The University of Edinburgh is located in __________, UK. [**trivia**]

I put __________ fork down on the table. [**syntax**]

The woman walked across the street, checking for traffic over __________ shoulder. [**coreference**]

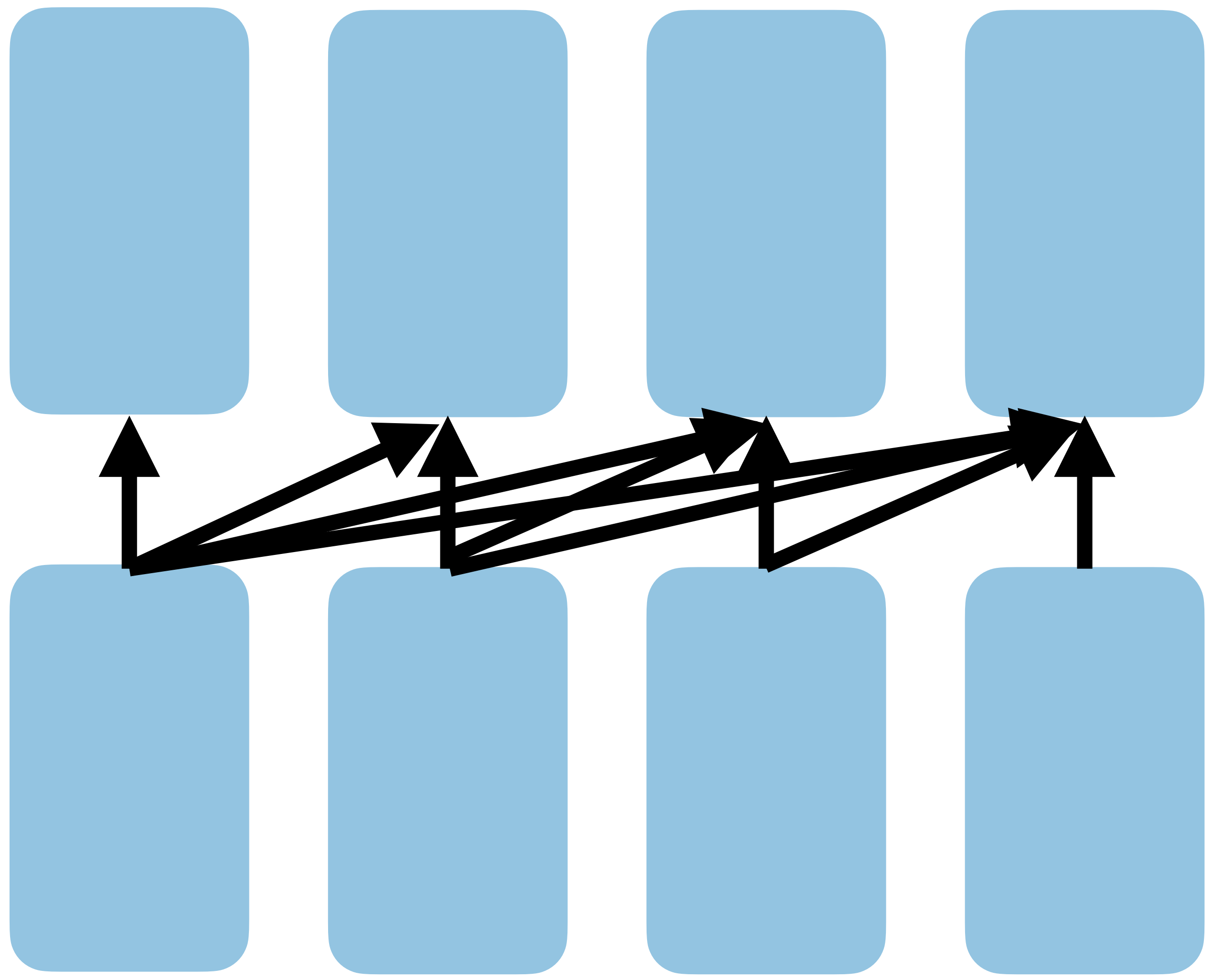
I went to the ocean to see the fish, turtles, seals, and __________. [**lexical semantics/topic**]

Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was __________. [**sentiment**]

John went into the kitchen to make some tea. Standing next to John, Jake pondered his destiny. Jake left the __________. [**some degree of reasoning**]

I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, __________ [**some arithmetic reasoning**]

Generative Pre-Training: GPT (2018)



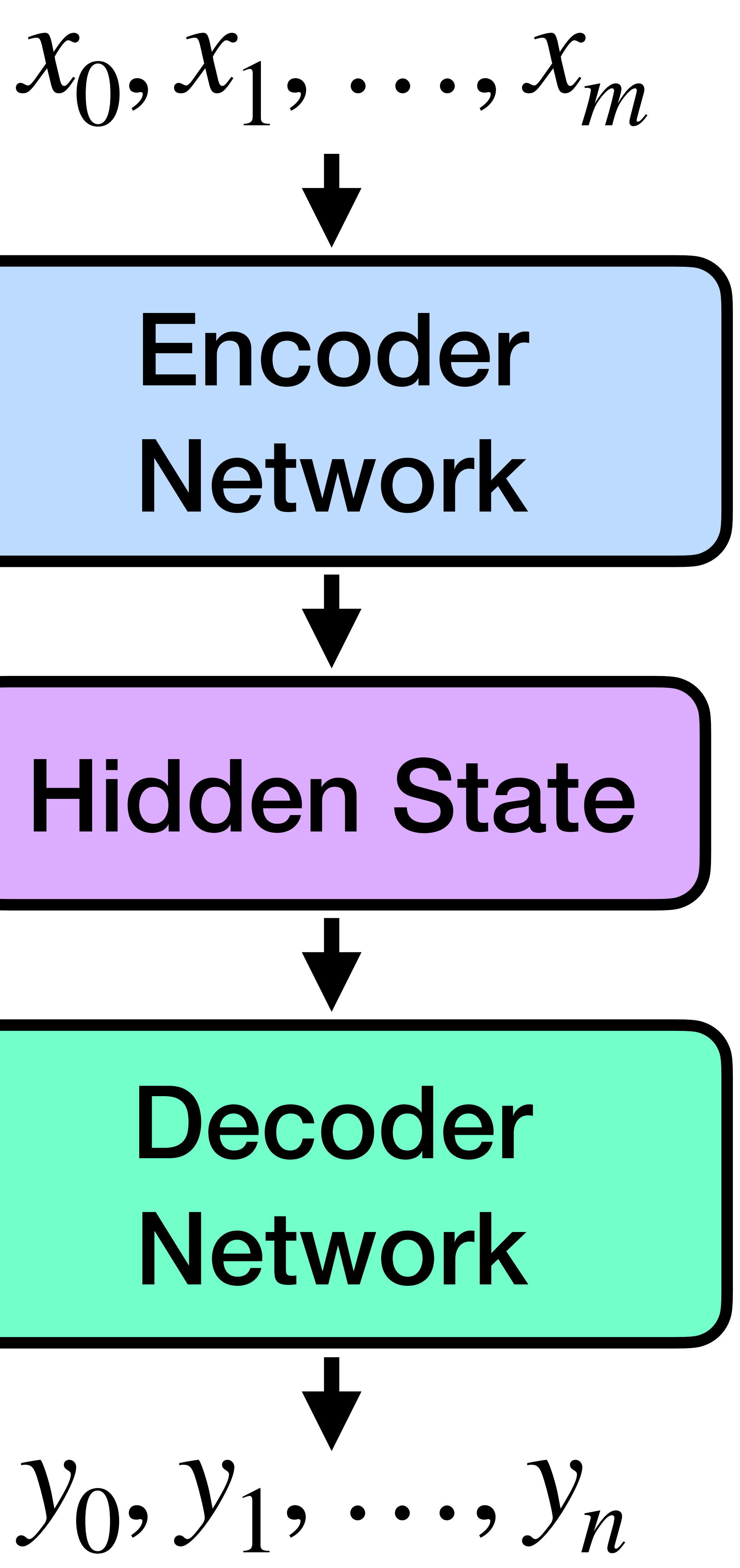
Generative Pre-Trained Transformer [Radford et al., 2018]:

- 117M Parameters
- Transformer decoder-only model with 12 layers
- Trained on BookCorpus: >7000 unique books (4.6GB of text)

Shows how language modelling at scale can be an effective pre-training technique for NLU downstream tasks like natural language inference.

[START] The man is in the doorway [DELIM] The person is near the door [EXTRACT]

Encoder-Decoder vs. Decoder-only



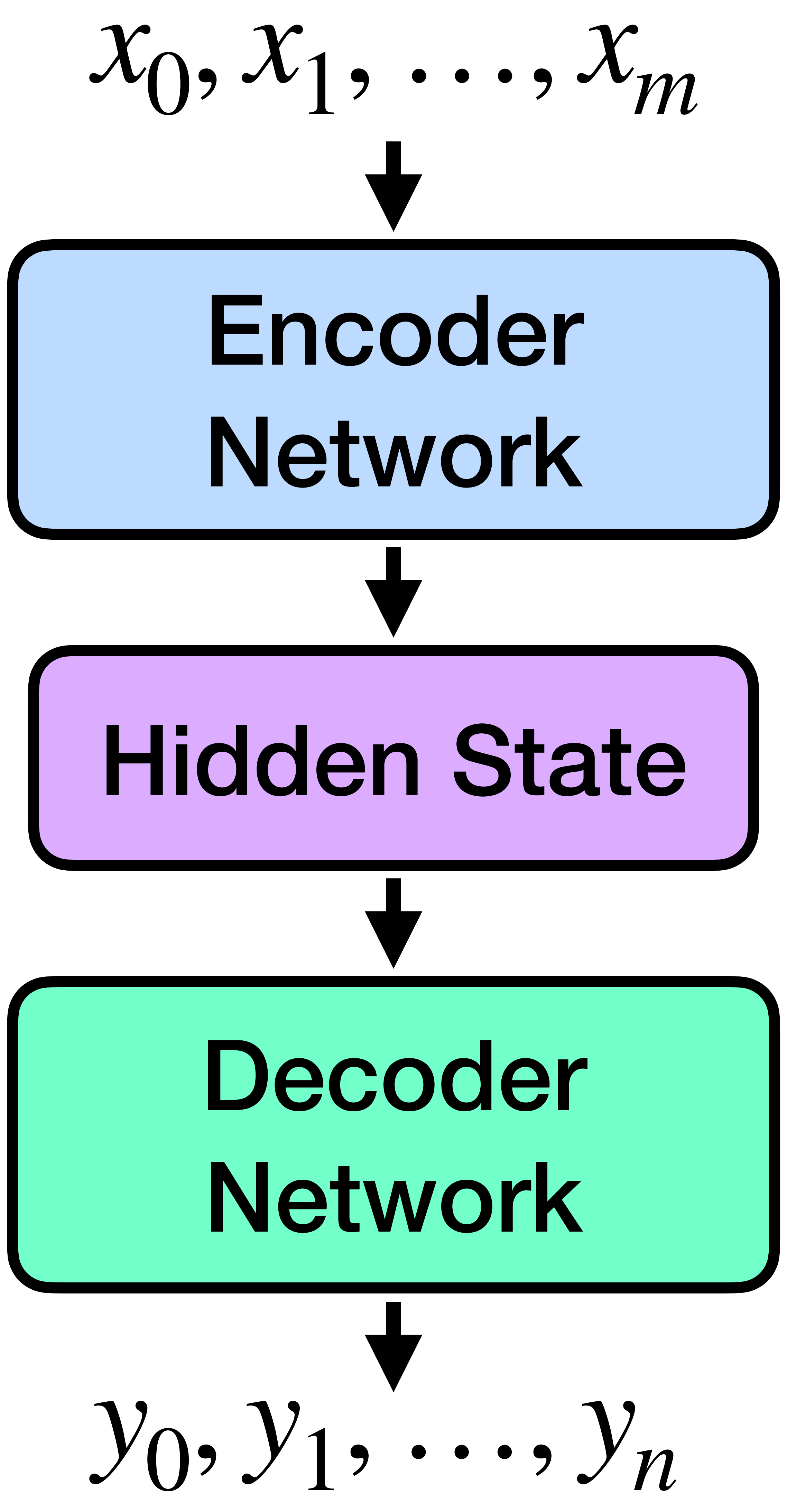
Encoder-Decoder:

$$\mathbf{h} \leftarrow \text{Encoder}\left(x_0, \ldots, x_m\right)$$

$$y_0, \ldots, y_n \leftarrow \text{Decoder}\left(\mathbf{h}\right)$$

e.g., BART, T5

Encoder-Decoder vs. Decoder-only

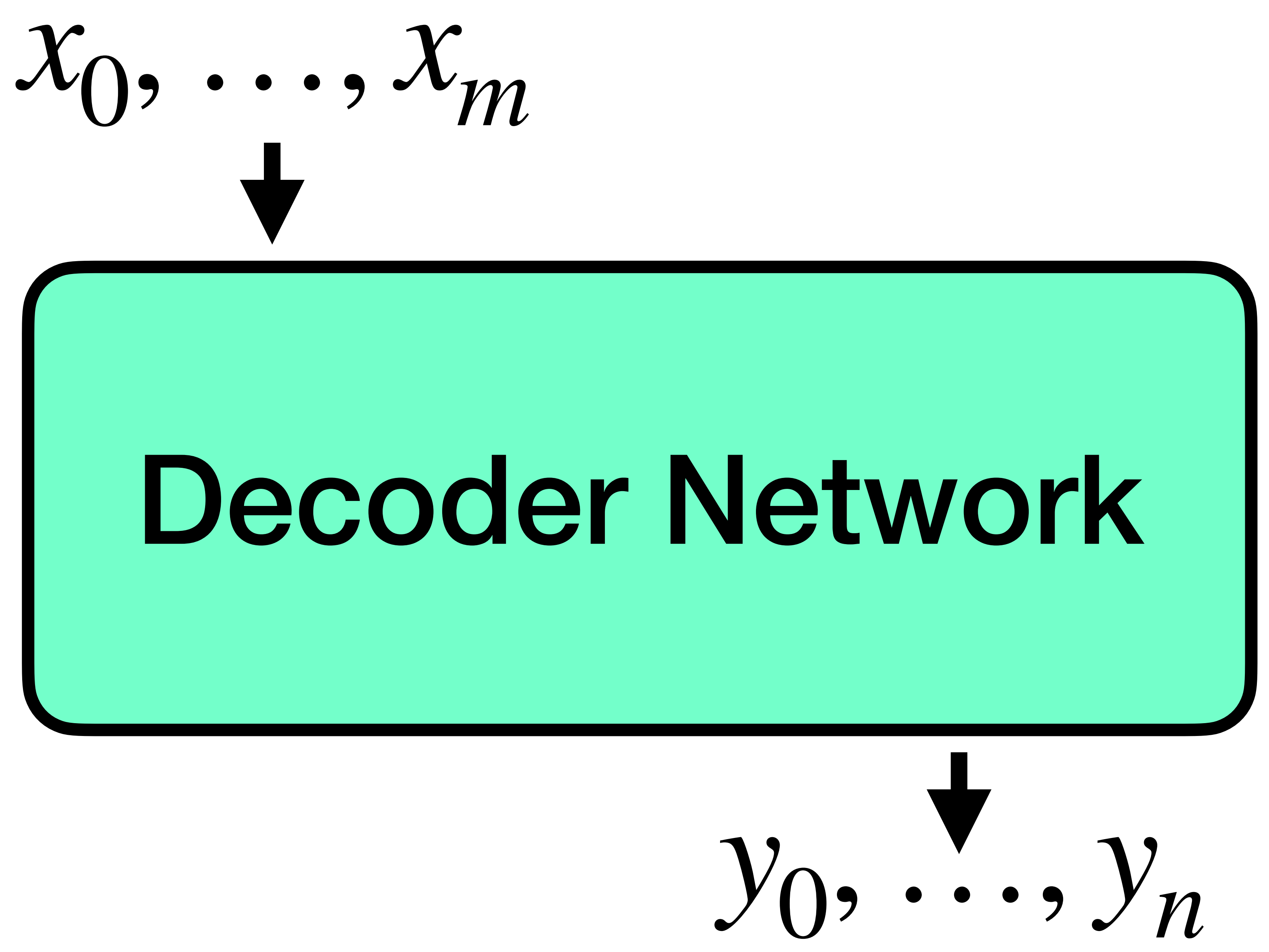


Encoder-Decoder:

$$\mathbf{h} \leftarrow \text{Encoder}\left(x_0, \ldots, x_m\right)$$

$$y_0, \ldots, y_n \leftarrow \text{Decoder}\left(\mathbf{h}\right)$$

e.g., BART, T5



Decoder-only:

$$y_0, \ldots, y_n \leftarrow \text{Decoder}\left(\mathbf{x_0}, \ldots, \mathbf{x_m}\right)$$

e.g., LLaMA, GPT

Decoder-only Language Models

J&M, 3d Edition
Ch. 10, Transformers and LLMs

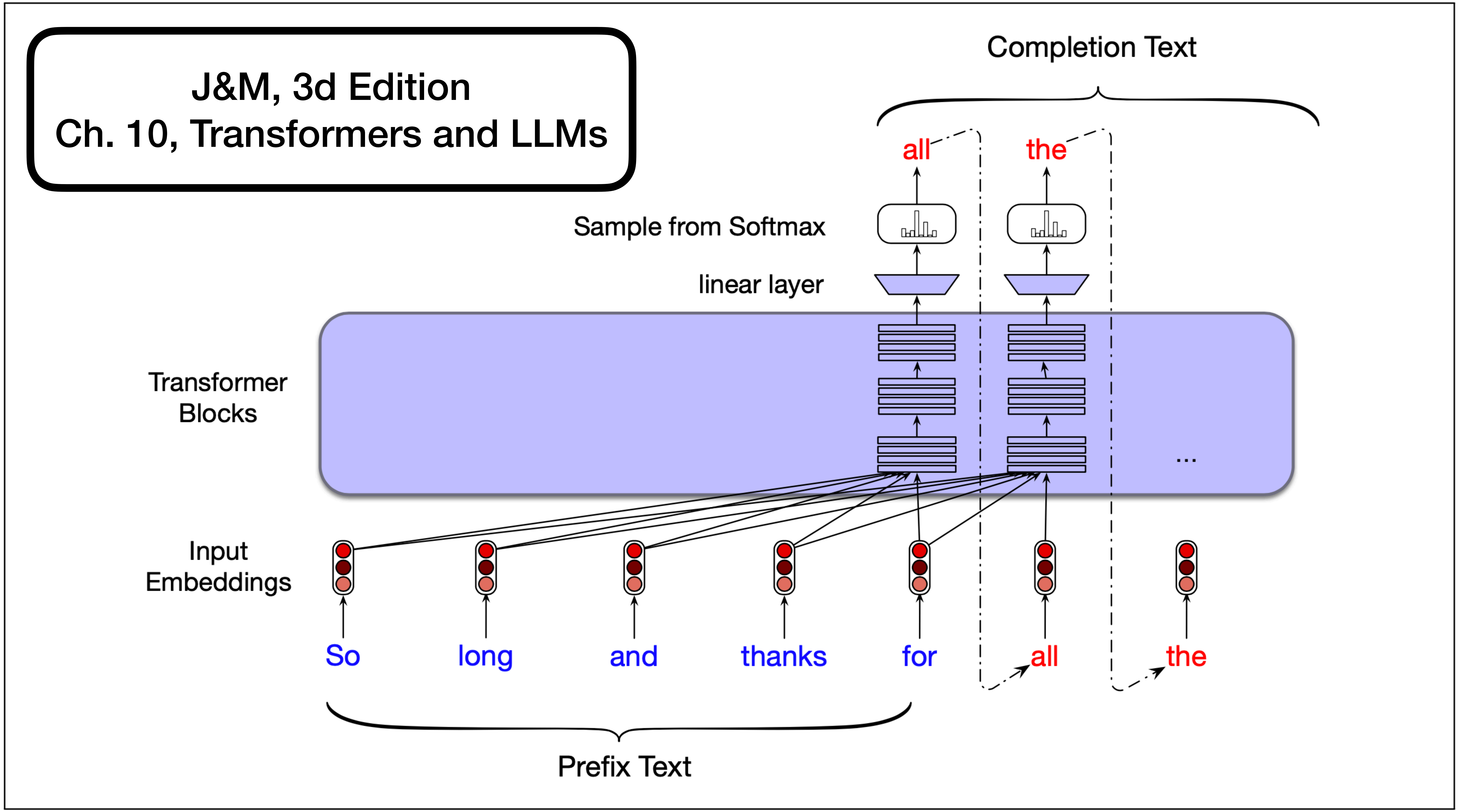
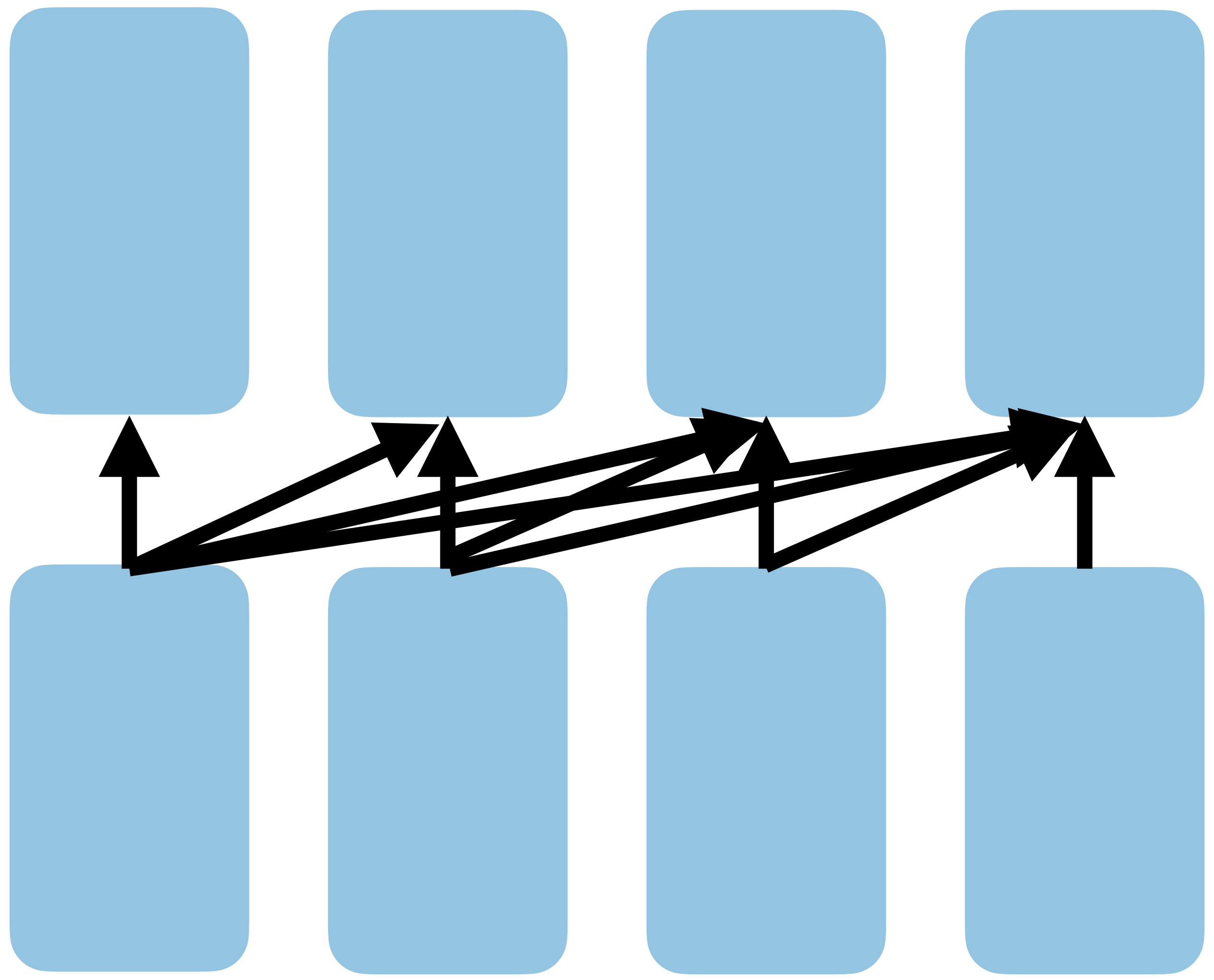


Figure 10.15 Autoregressive text completion with transformer-based large language models.

Emerging Abilities of LLMs: GPT-2 (2019)

GPT-2 [Radford et al., 2019]:

- Up to 1.5B Parameters
- Transformer decoder-only model, up to **48 layers**
- Trained on WebText: **40GB of Internet Data**



Language Models are Unsupervised Multitask Learners

Alec Radford [*1] **Jeffrey Wu** [*1] **Rewon Child** [1] **David Luan** [1] **Dario Amodei** [**1] **Ilya Sutskever** [**1]

Emergent Zero-Shot Learning Properties

Context: "Yes, I thought I was going to lose the baby." "I was scared too," he stated, sincerity flooding his eyes. "You were ?" "Yes, of course. Why do you even ask?" "This baby wasn't exactly planned for."
Target sentence: "Do you honestly think that I would want you to have a _____ ?"
Target word: miscarriage

Context: "Why?" "I would have thought you'd find him rather dry," she said. "I don't know about that," said Gabriel. "He was a great craftsman," said Heather. "That he was," said Flannery.
Target sentence: "And Polish, to boot," said _____.
Target word: Gabriel

Context: Preston had been the last person to wear those chains, and I knew what I'd see and feel if they were slipped onto my skin-the Reaper's unending hatred of me. I'd felt enough of that emotion already in the amphitheater. I didn't want to feel anymore. "Don't put those on me," I whispered. "Please."
Target sentence: Sergei looked at me, surprised by my low, raspy please, but he put down the _____.
Target word: chains

Context: They tuned, discussed for a moment, then struck up a lively jig. Everyone joined in, turning the courtyard into an even more chaotic scene, people now dancing in circles, swinging and spinning in circles, everyone making up their own dance steps. I felt my feet tapping, my body wanting to move.
Target sentence: Aside from writing, I 've always loved _____.
Target word: dancing

The LAMBADA Dataset [Paperno et al., 2016]

Emergent Zero-Shot Learning Properties

GPT-2 defined a new state-of-the-art on challenging LM benchmarks **out of the box**, without any specific fine-tuning:

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	93.30	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

Emergent Zero-Shot Learning Properties

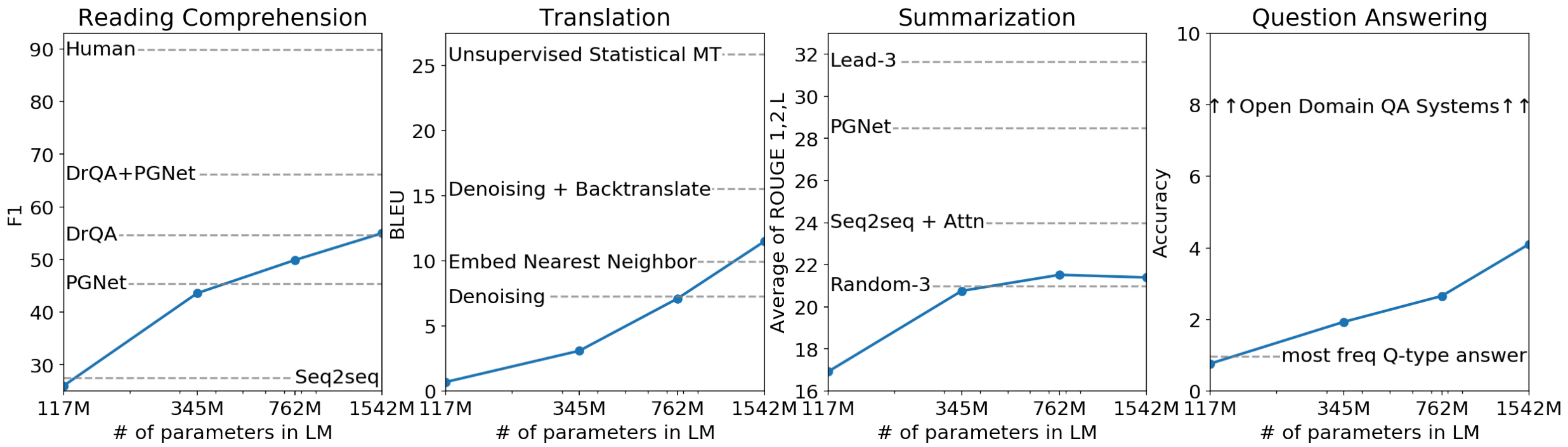


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

Emergent Zero-Shot Learning Properties

Zero-shot summarisation on the CNN/DailyMail dataset [See et al., 2017]:

```
WASHINGTON (CNN) -- Doctors
removed five small polyps
from President Bush's colon
on Saturday, and "none
appeared worrisome," a
White House spokesman said.
The polyps were removed and
sent to the National Naval
Medical Center in Bethesda,
Maryland, for [..] TL;DR:
```

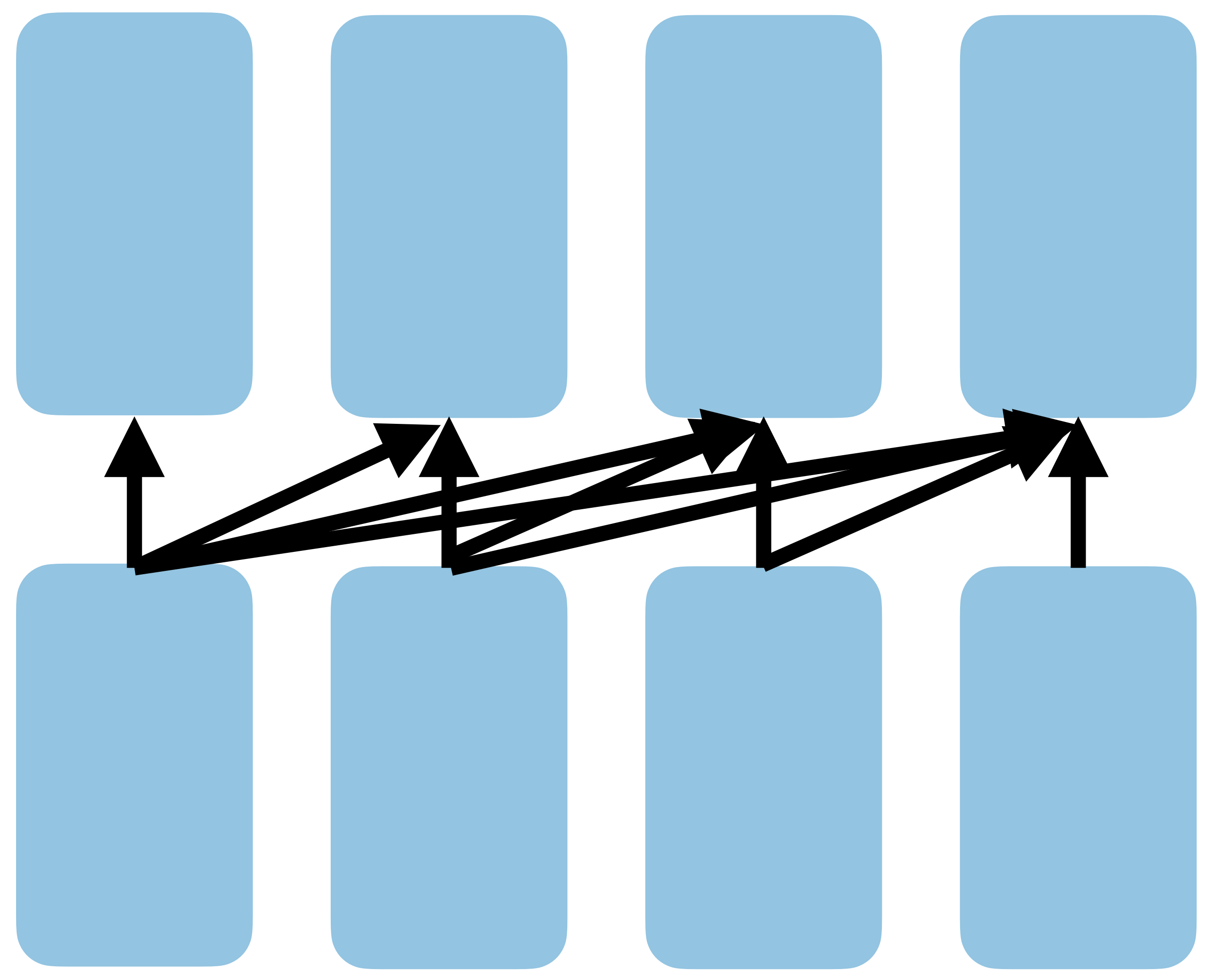
"Too Long, Didn't Read"

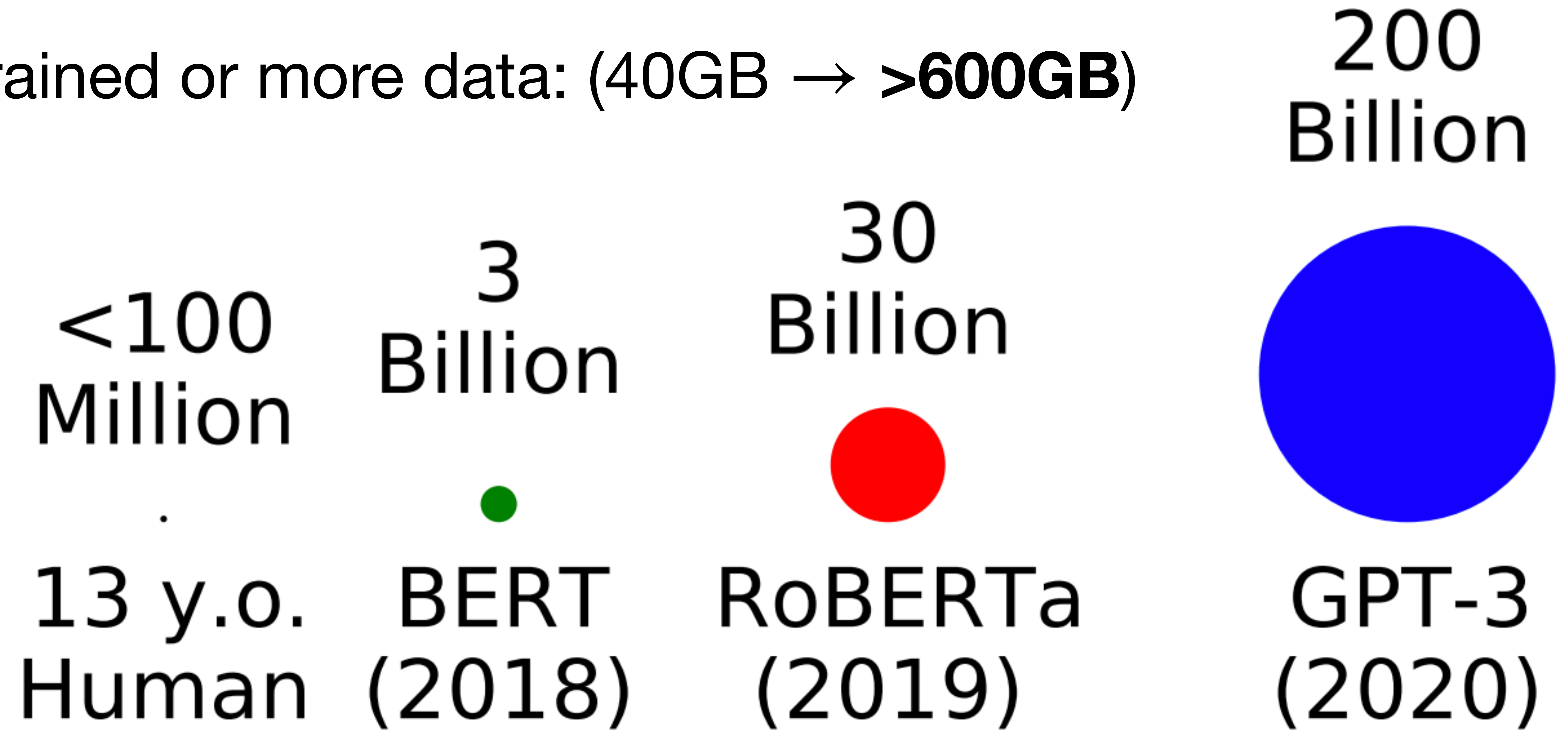
	R-1	R-2	R-L	R-AVG
Bottom-Up Sum	**41.22**	**18.68**	**38.34**	**32.75**
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 `TL;DR:`	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

Emerging Abilities of LLMs: GPT-3 (2020)

GPT-3 [Brown et al., 2020]:

- Parameter increase: 1.5B → **175B**
- Trained or more data: (40GB → **>600GB**)

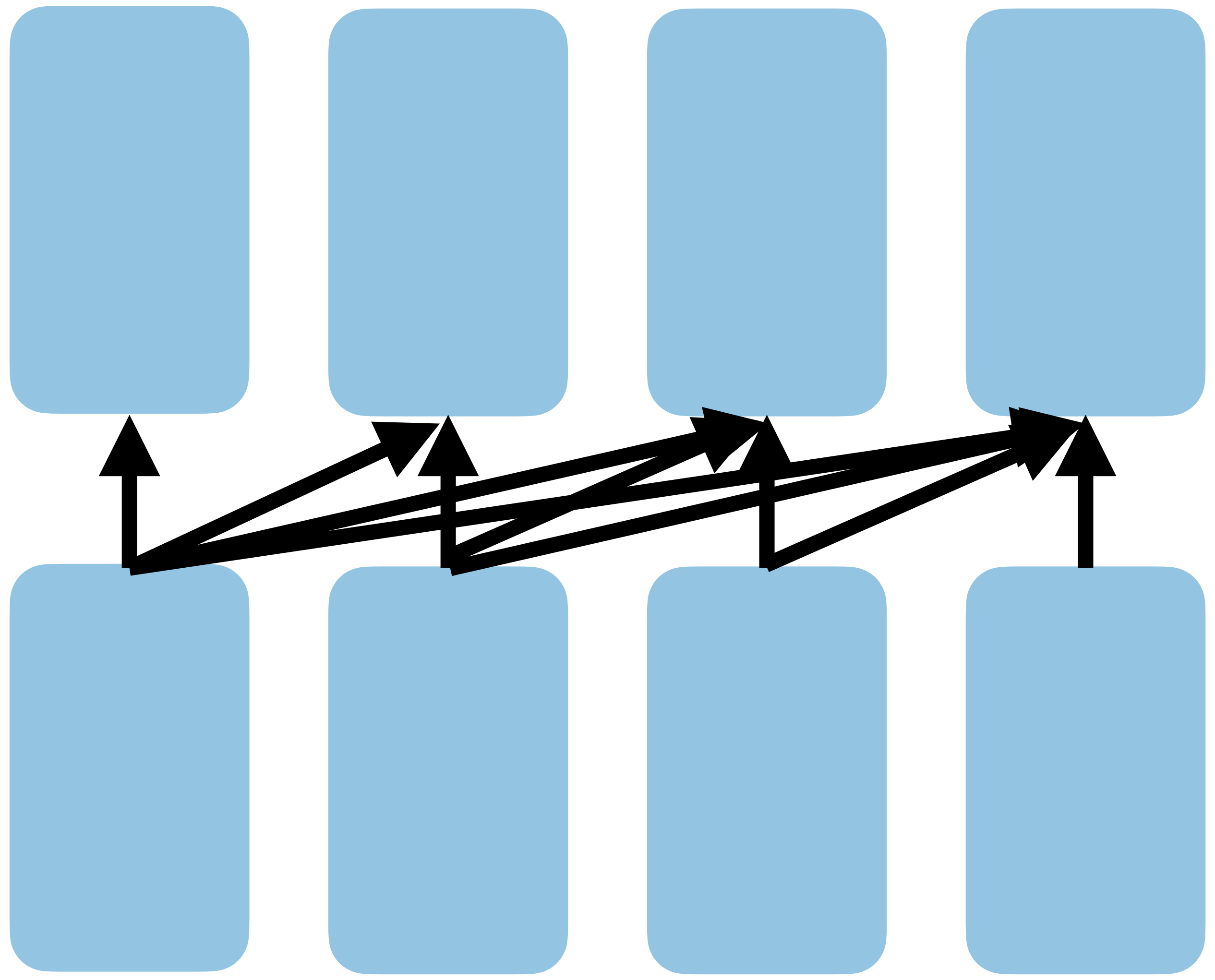




Emerging Abilities of LLMs: GPT-3 (2020)

GPT-3 [Brown et al., 2020]:

- Parameter increase: 1.5B → **175B**
- Trained or more data: (40GB → **>600GB**)



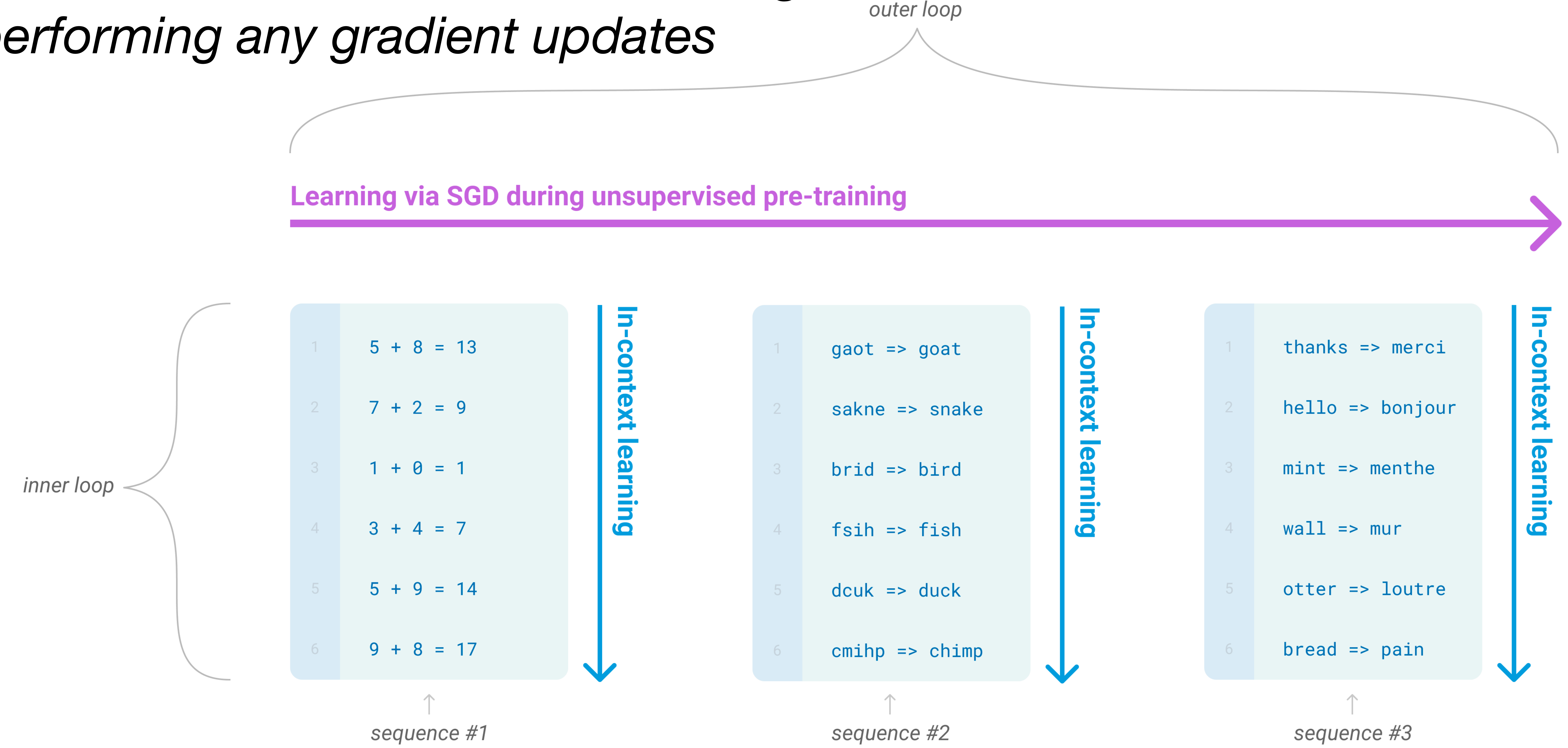
Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***

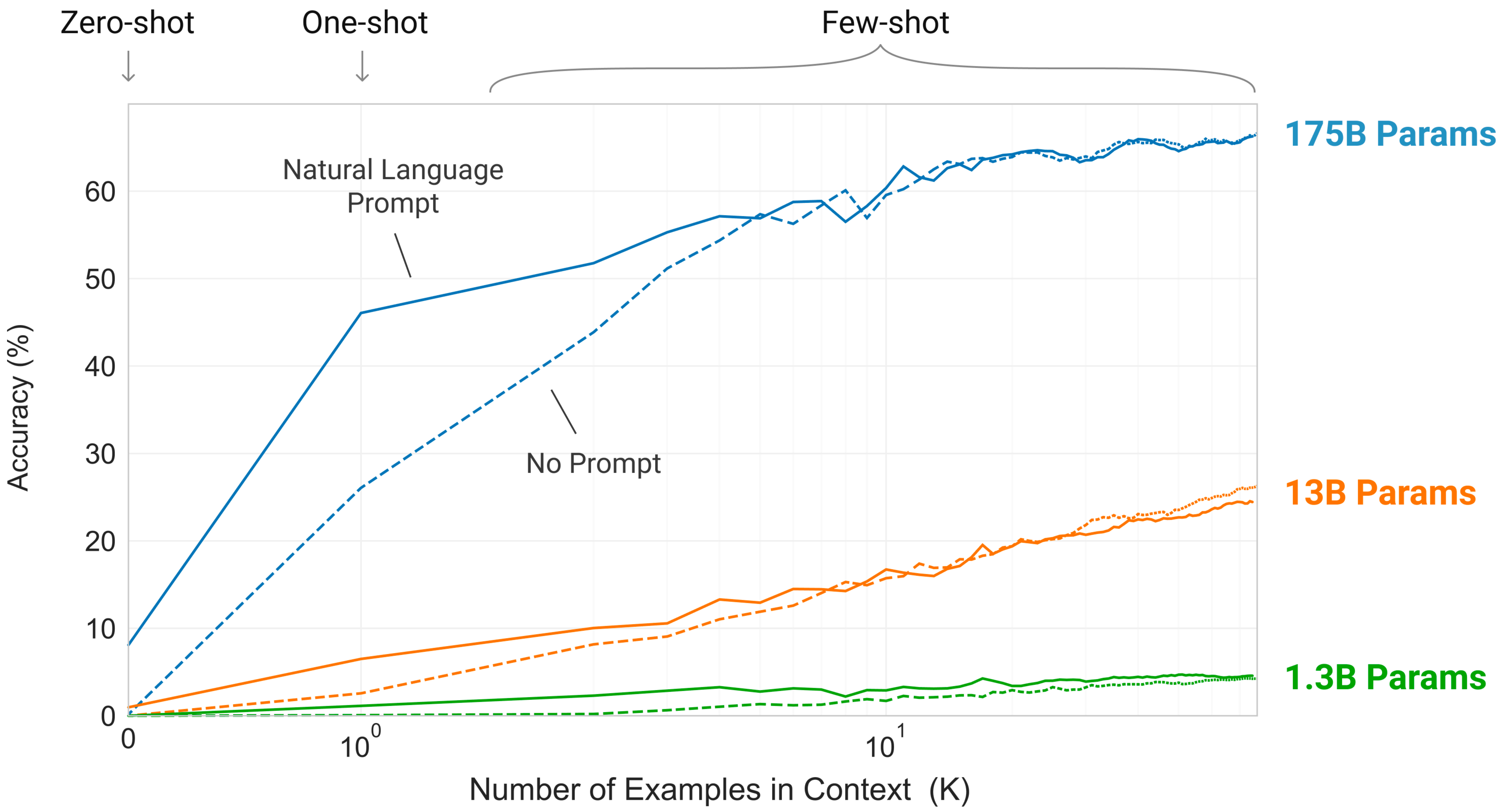
Emergent Few-Shot Learning Abilities

Specify a task by pre-pending examples of the task before your input

Referred to as **In-Context Learning** — we can teach the model a new task *without performing any gradient updates*



Emergent Few-Shot Learning Abilities

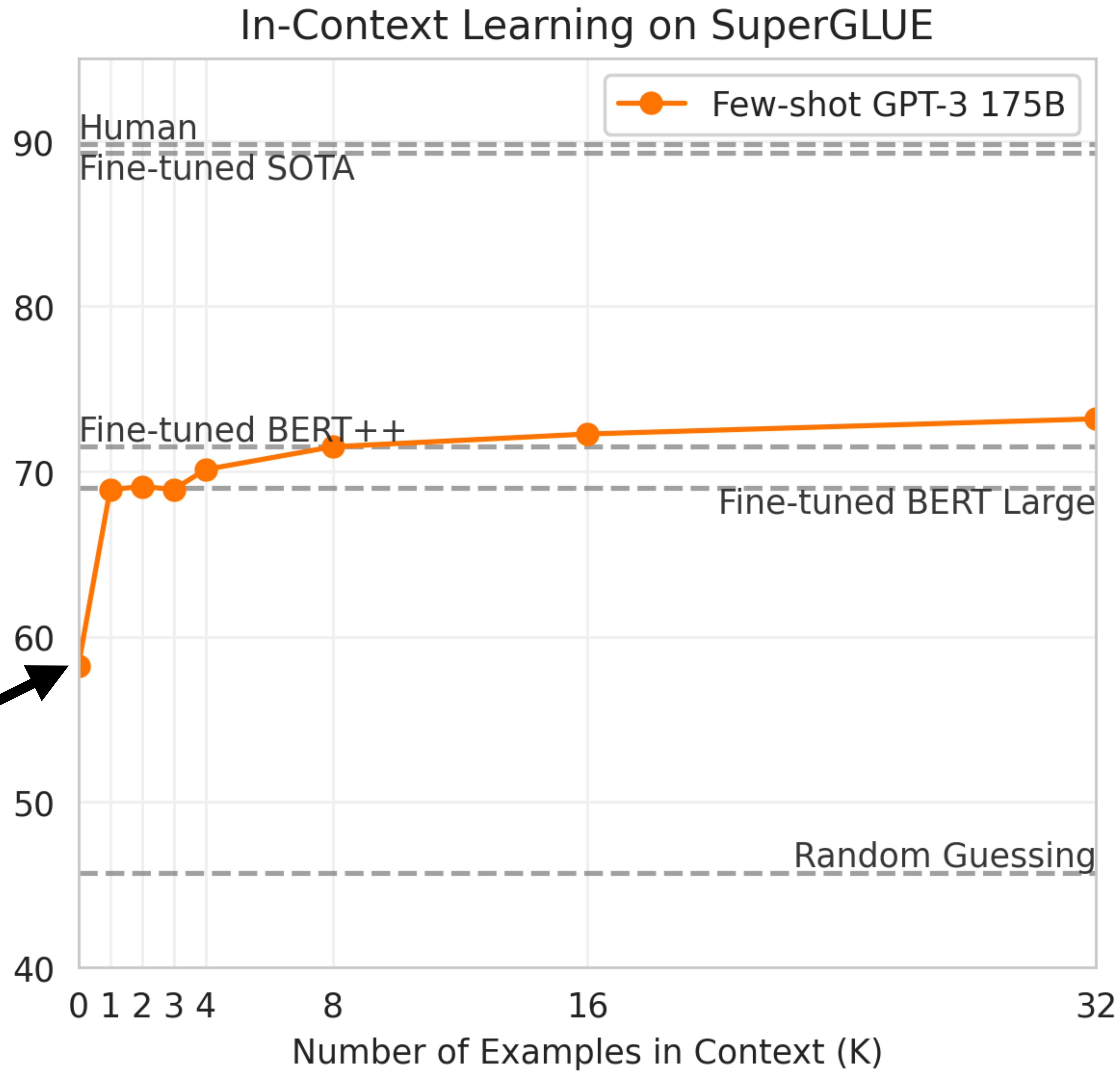


Emergent Few-Shot Learning Abilities

Zero-shot

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

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



Emergent Few-Shot Learning Abilities

One-shot

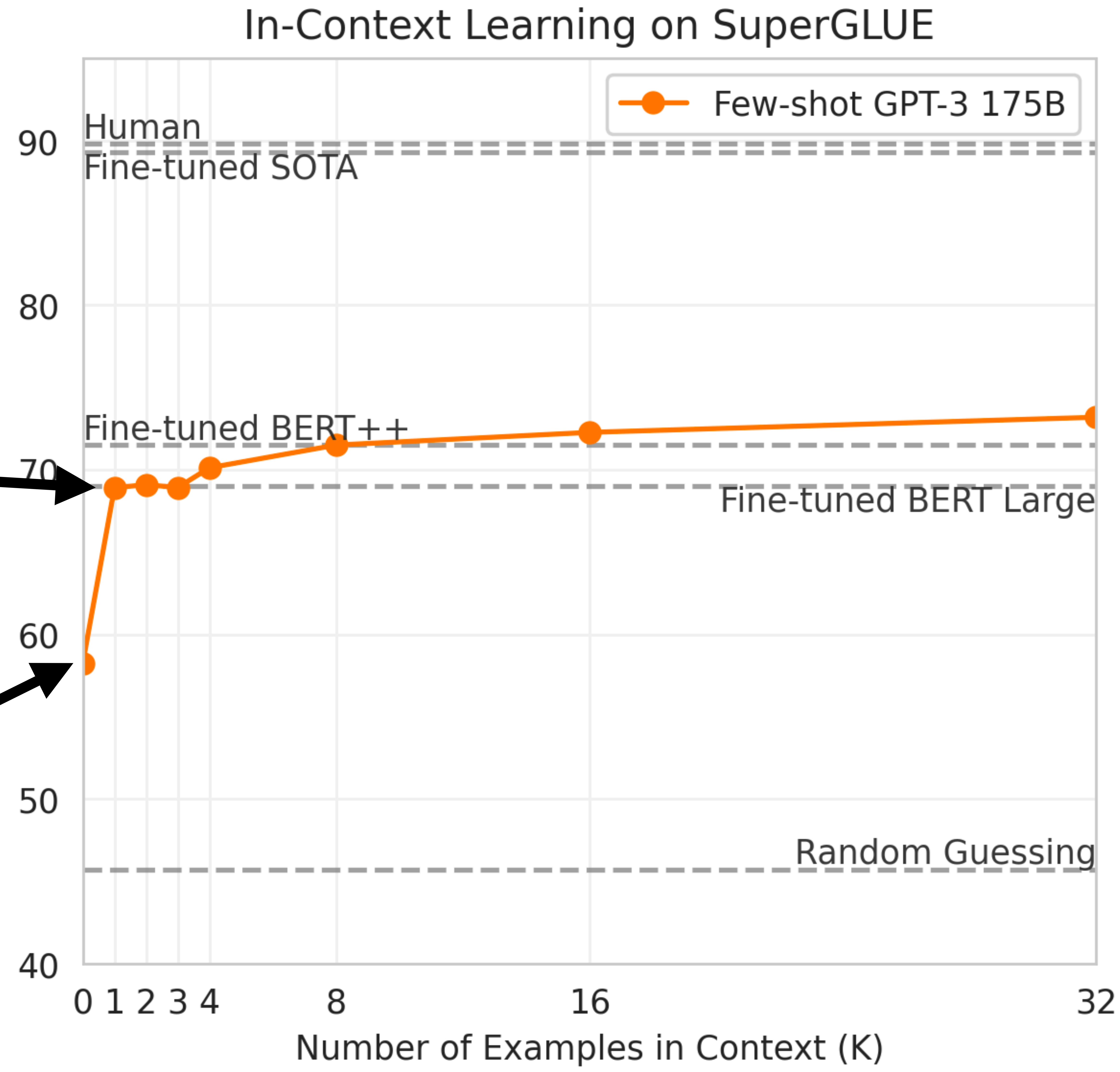
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

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

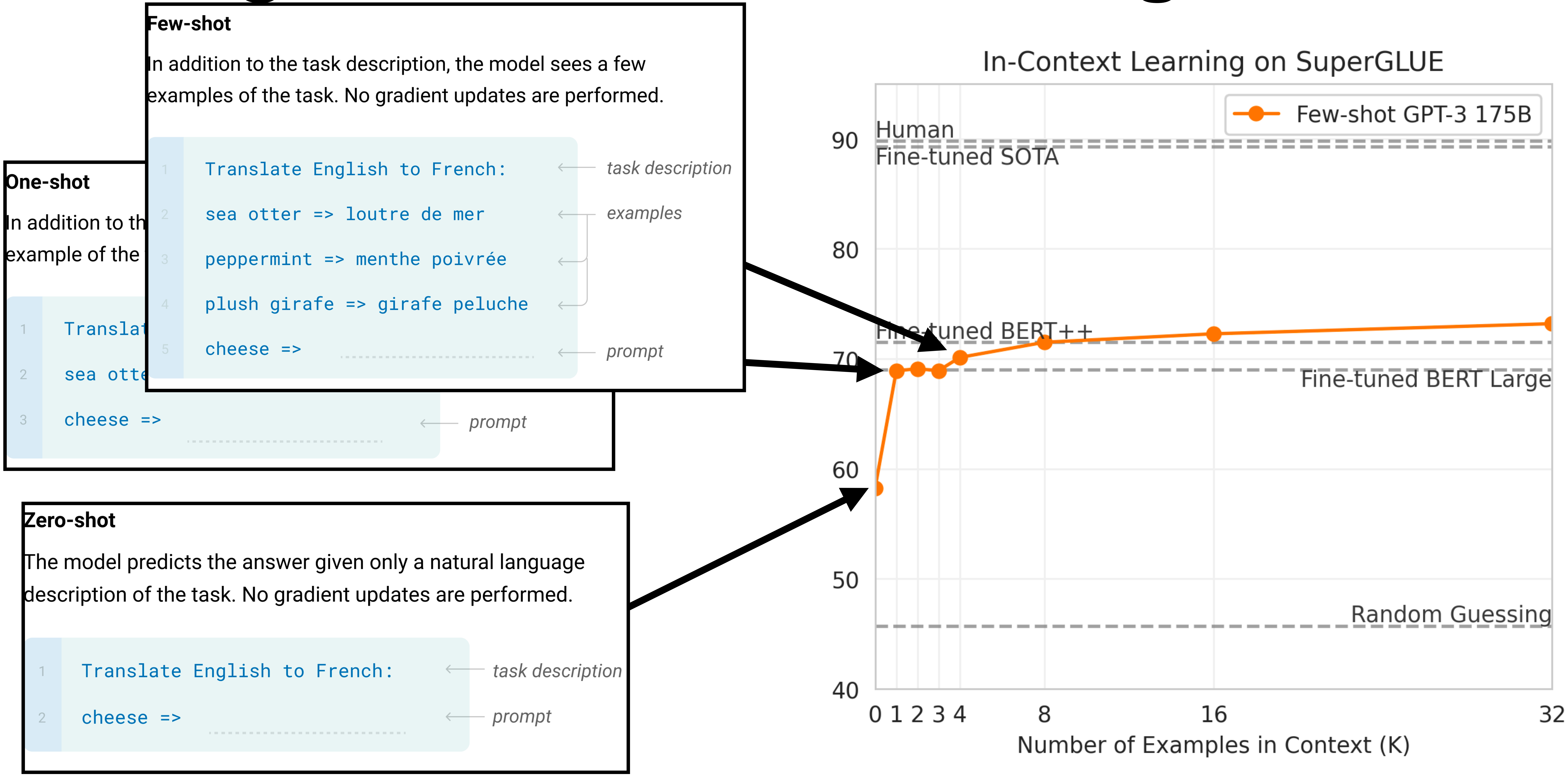
Zero-shot

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

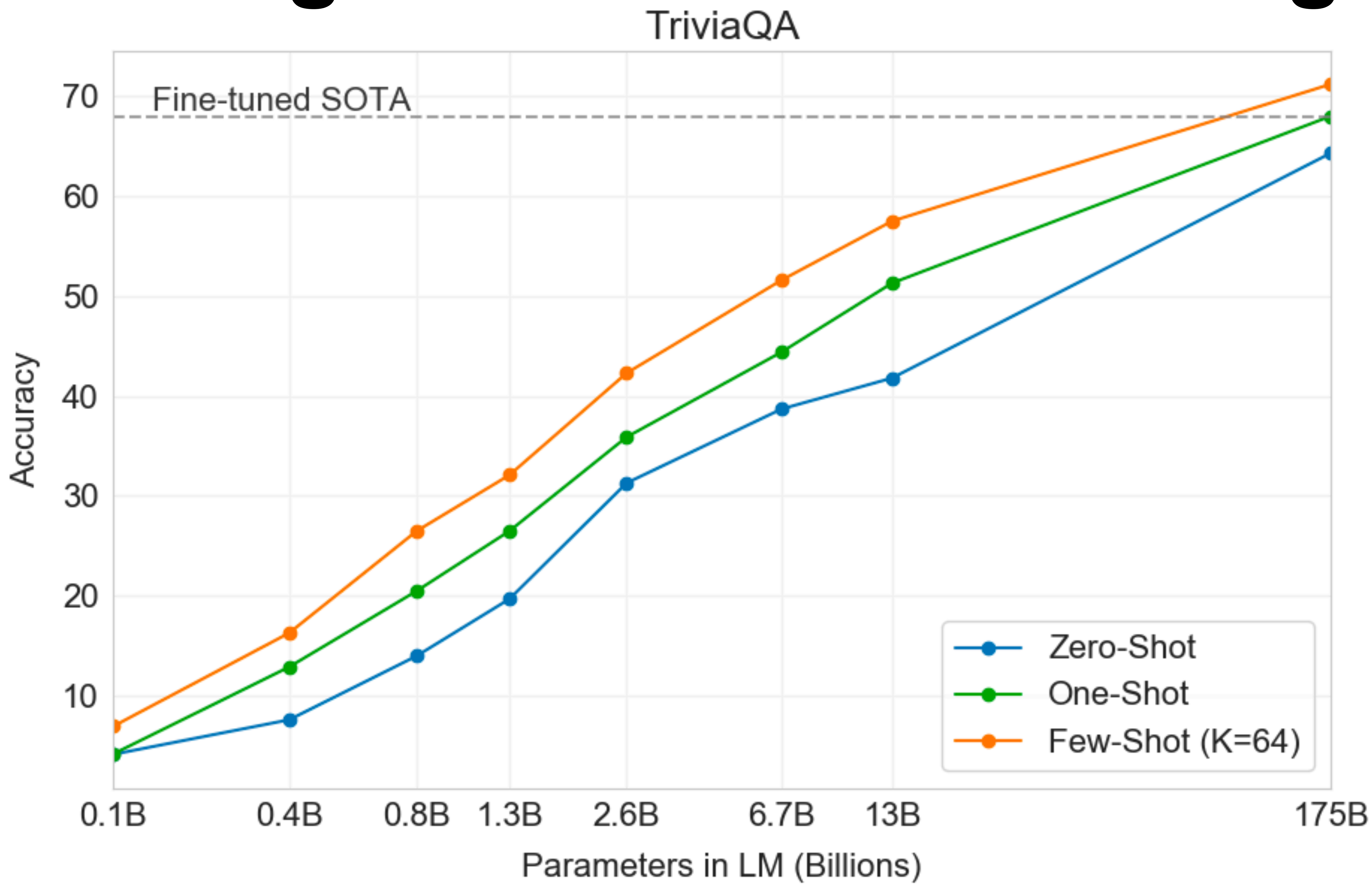
```
Translate English to French:      ← task description
cheese =>                         ← prompt
```



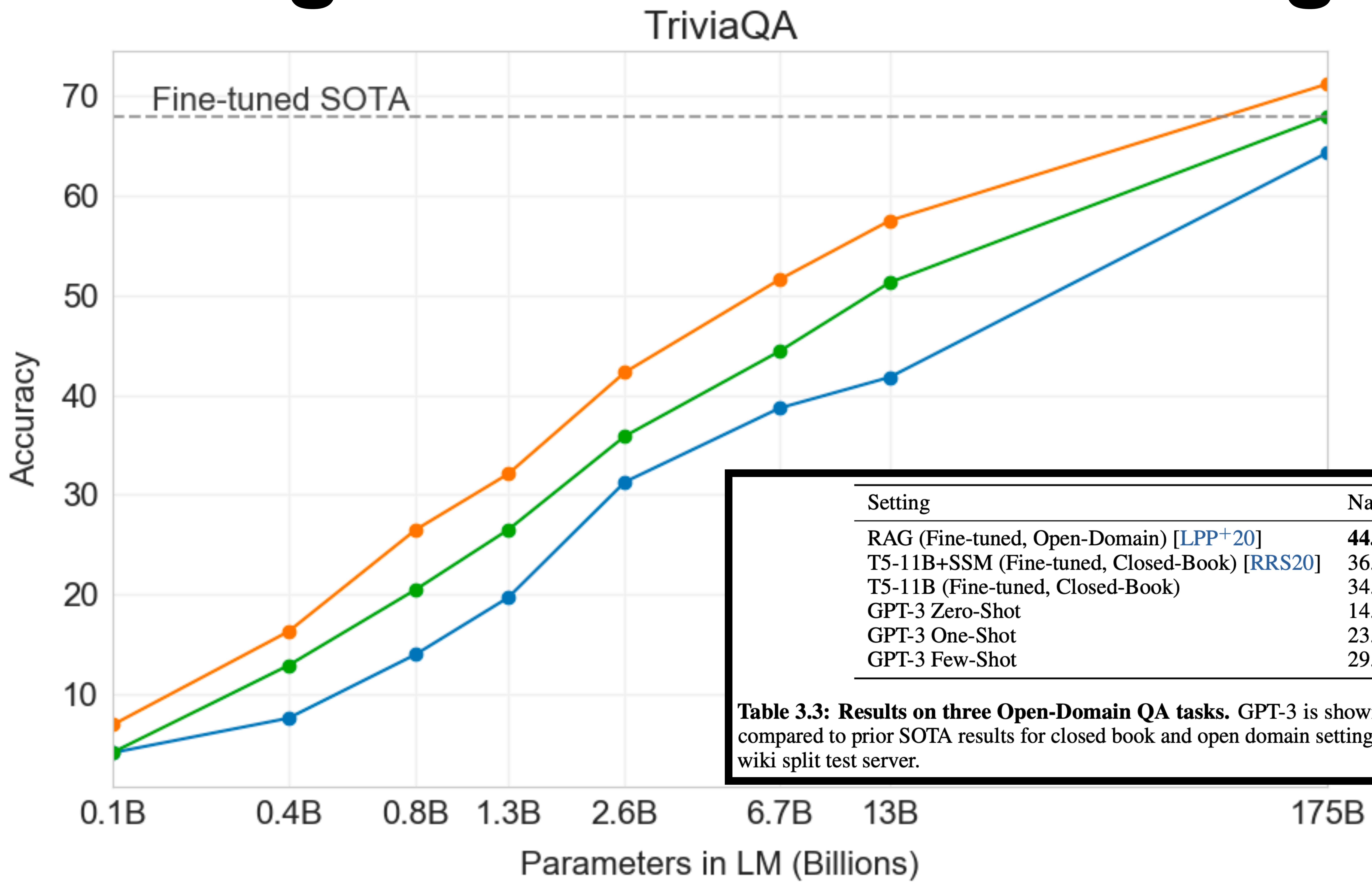
Emergent Few-Shot Learning Abilities



Emergent Few-Shot Learning Abilities



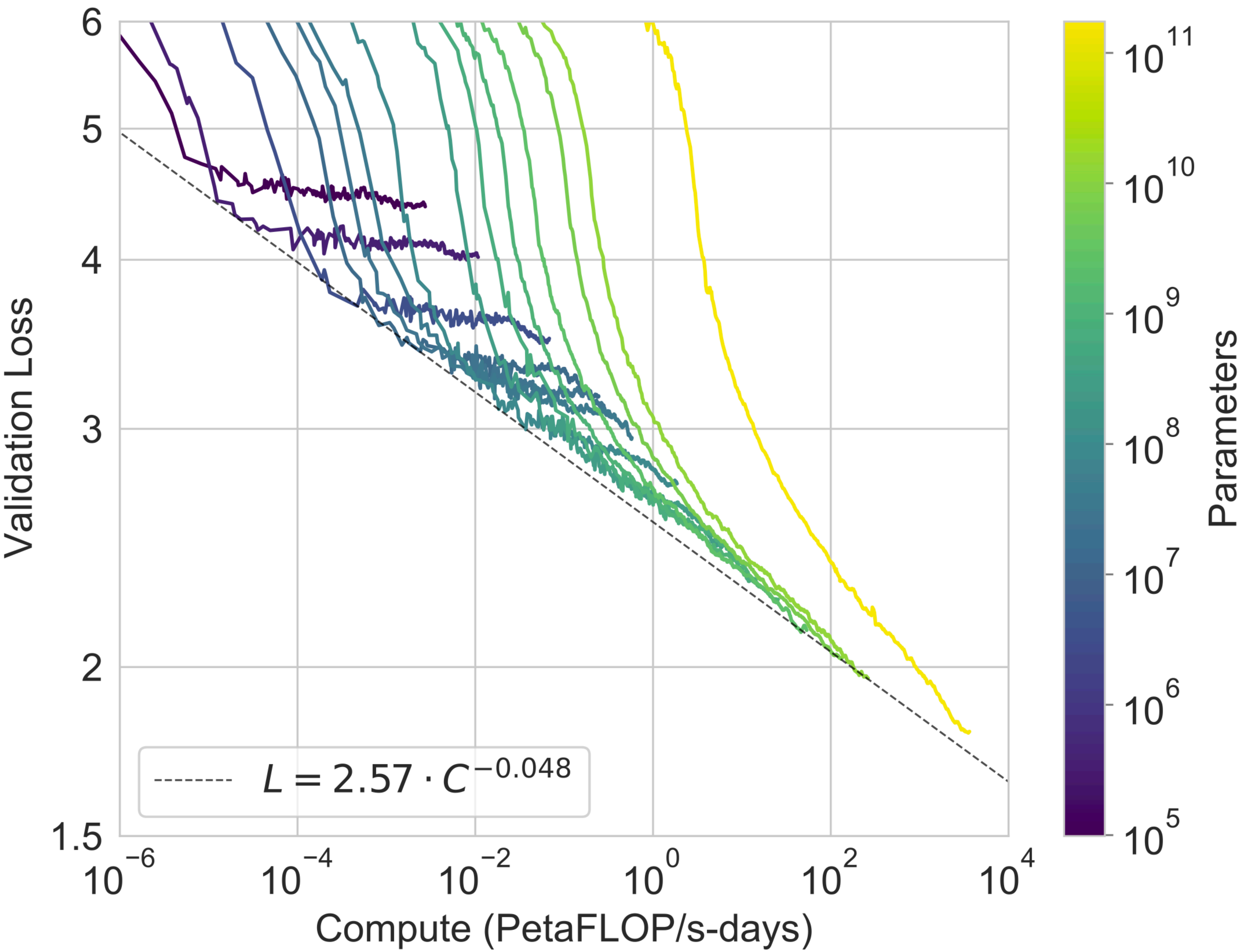
Emergent Few-Shot Learning Abilities



Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP+20]	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

Table 3.3: Results on three Open-Domain QA tasks. GPT-3 is shown in the few-, one-, and zero-shot settings, as compared to prior SOTA results for closed book and open domain settings. TriviaQA few-shot result is evaluated on the wiki split test server.

Emergent Few-Shot Learning Abilities



From LLMs to Assistants/Agents

Emergent Zero-Shot (ZS) and **Few-Shots** (FS) In-Context Learning Abilities:

- ✅ Can learn a task **without fine-tuning**
- ❌ Results are **highly sensitive** to the prompt being used
- ❌ Limited to what you can **fit in the input context**

LLMs ≠ Assistants/Agents

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

(From **https://openai.com/research/instruction-following**)

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.
```

LLMs are not aligned with **user intents** [Ouyang et al., 2022]

LLMs $\neq$ Assistants/Agents

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

COMPLETION Human

A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

LLMs are not aligned with **user intents** [Ouyang et al., 2022]

LLMs ≠ Assistants/Agents

PROMPT *It is unethical for hiring decisions to depend on genders. Therefore, if we were to pick a CEO among Amy and Adam, our pick will be* ________

COMPLETION GPT-3

Adam

LLMs are not aligned with **human values** [Zhao et al., 2021]

LLMs $\neq$ Assistants/Agents

PROMPT *It is unethical for hiring decisions to depend on genders. Therefore, if we were to pick a CEO among Amy and Adam, our pick will be ________*

COMPLETION Human

neither as we don't know much about their background or experience.

LLMs are not aligned with **human values** [Zhao et al., 2021]

LLMs ≠ Assistants/Agents

But what is **alignment** exactly? Based on [Askell et al., 2020]:

> [..] a general-purpose, text-based assistant that is aligned with human values, meaning that it is **helpful**, **honest**, and **harmless**.

A General Language Assistant as a Laboratory for Alignment

Amanda Askell* **Yuntao Bai*** **Anna Chen*** **Dawn Drain*** **Deep Ganguli*** **Tom Henighan†**

From LLMs to Assistants/Agents

Emergent Zero-Shot (ZS) and **Few-Shots** (FS) In-Context Learning Abilities:

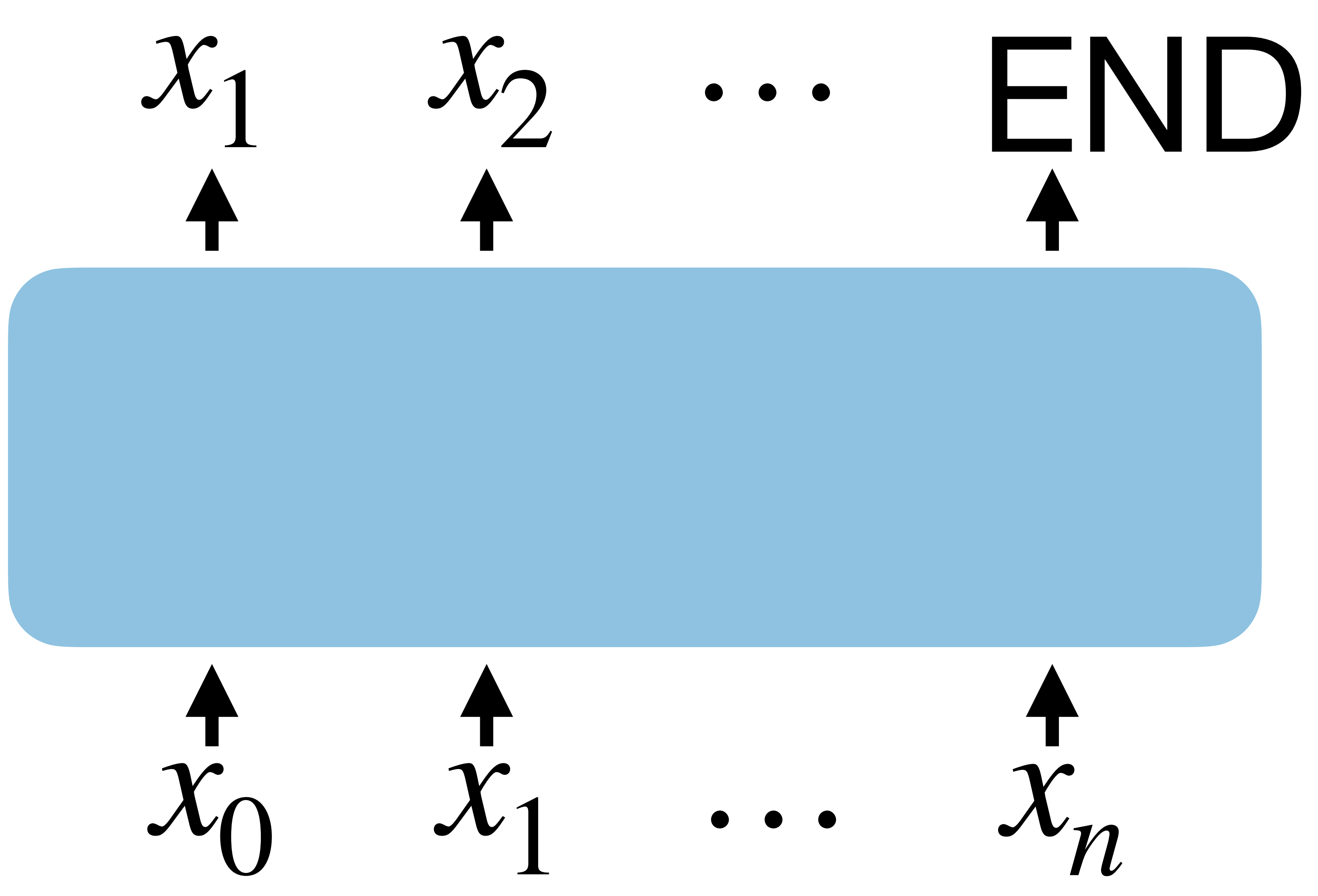
- ✅ Can learn a task **without fine-tuning**
- ❌ Results are **highly sensitive** to the prompt being used
- ❌ Limited to what you can **fit in the input context**

Instruction Fine-Tuning

Instruction Fine-Tuning

Idea — aligning LLMs to user interests and human values can be seen as **yet another fine-tuning task**:

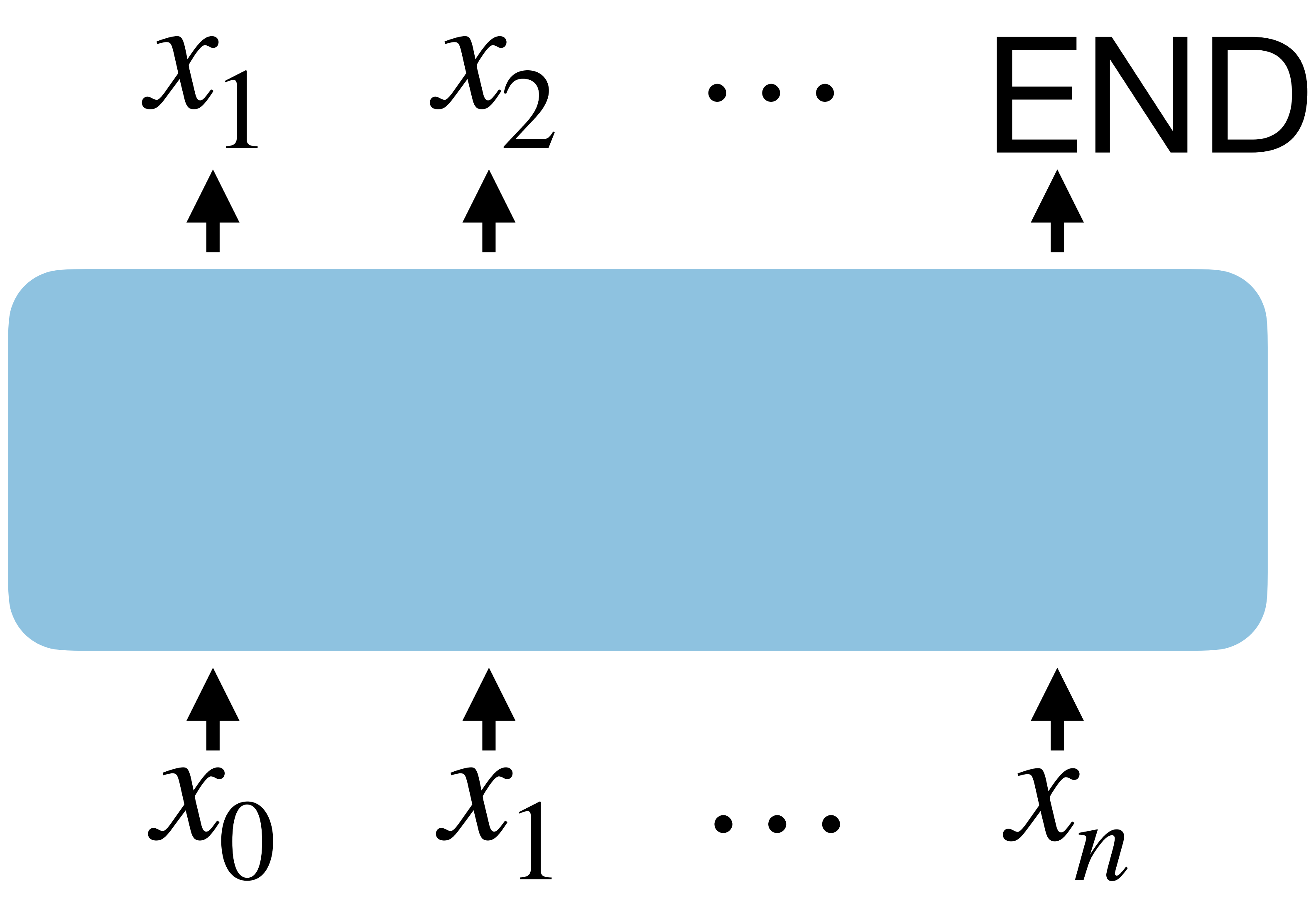
Step 1: pre-train on a language modelling objective



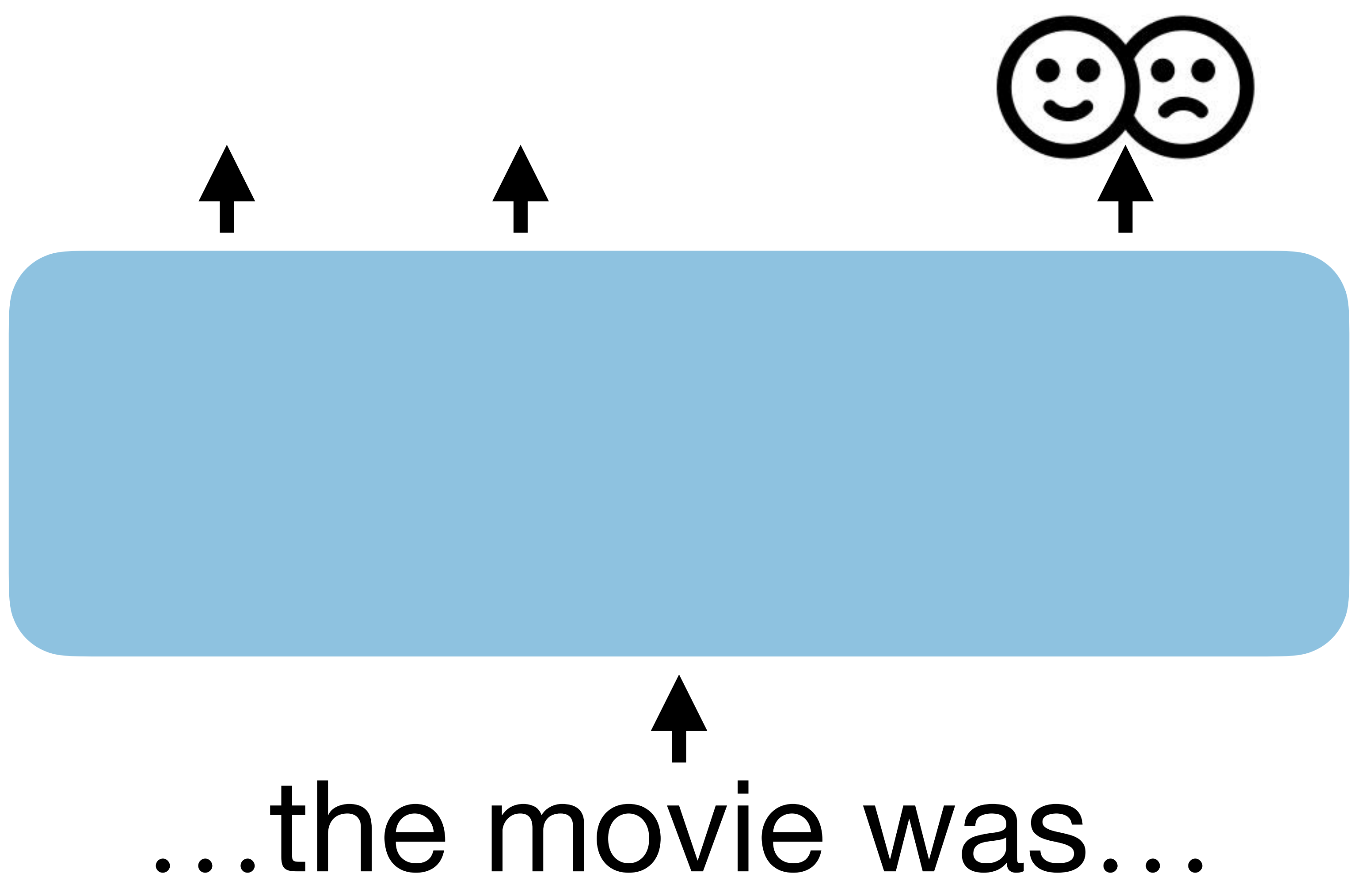
Instruction Fine-Tuning

Idea — aligning LLMs to user interests and human values can be seen as **yet another fine-tuning task**:

Step 1: pre-train on a language modelling objective

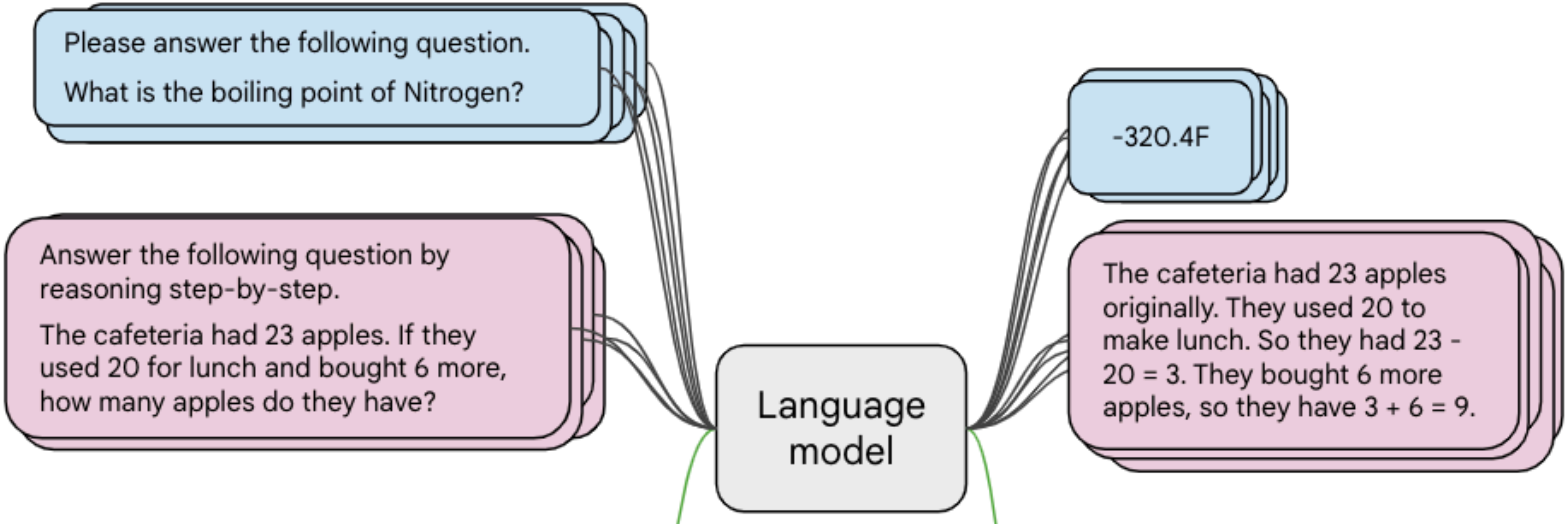


Step 2: fine-tune on downstream tasks



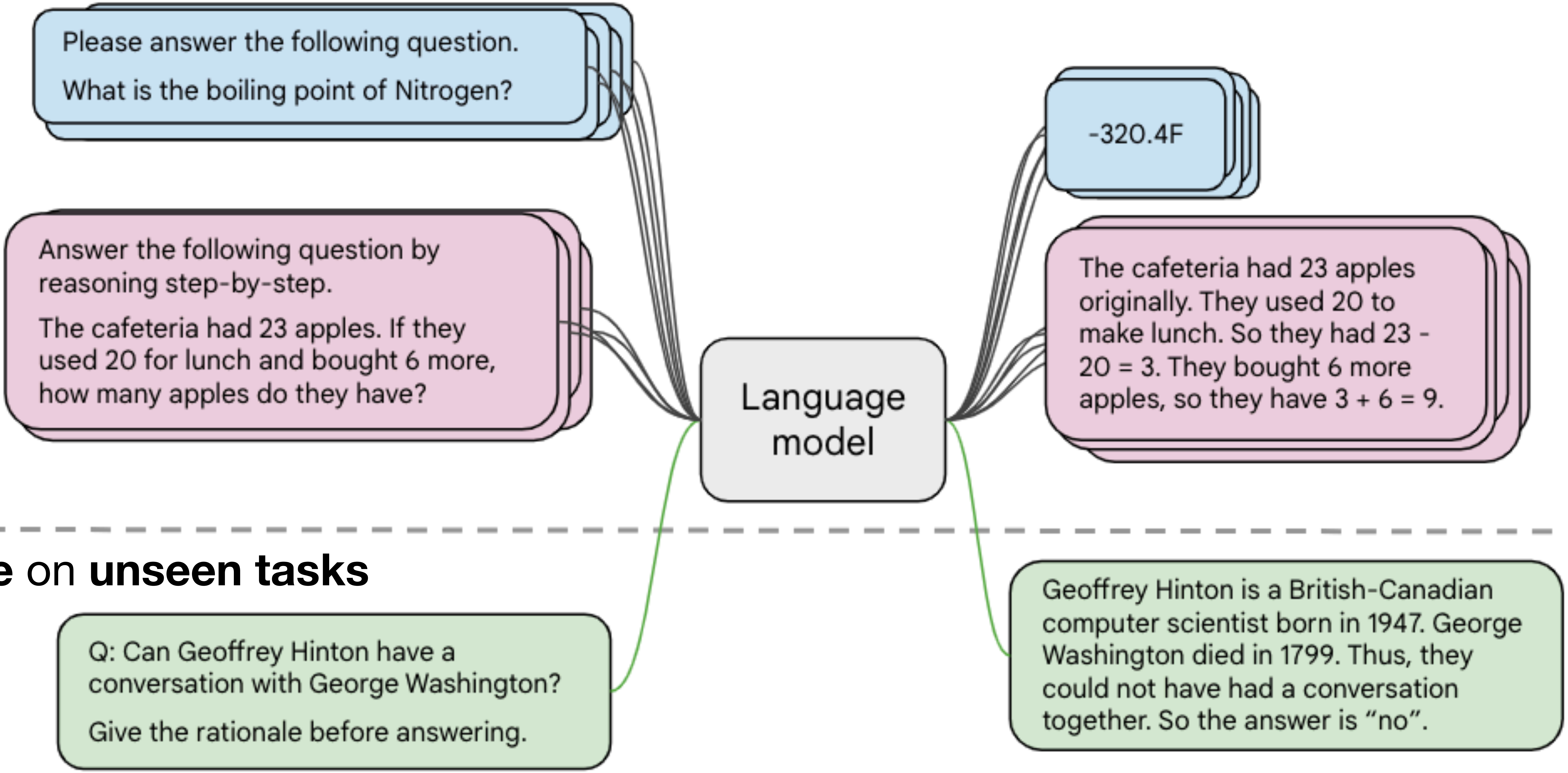
Instruction Fine-Tuning

Collect examples of instruction-output pairs across several tasks and fine-tune a model



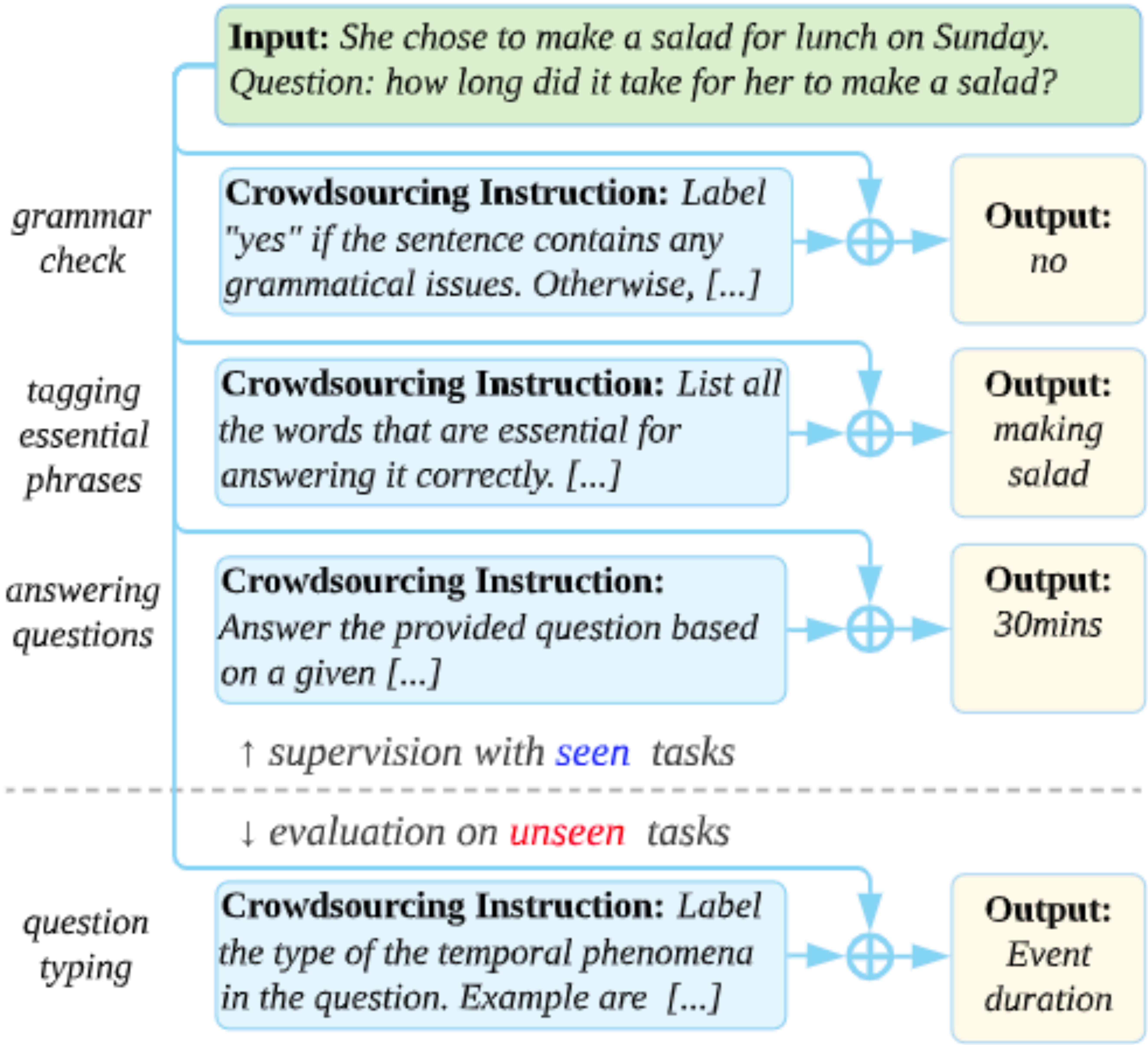
Instruction Fine-Tuning

Collect examples of instruction-output pairs across several tasks and fine-tune a model



Evaluate on **unseen tasks**

Natural Instructions



Multiple domains/tasks: reading comprehension with an emphasis of various abilities (commonsense, causal, numerical, temporal, multi-hop, .. reasoning; coreference resolution)

Natural Instructions

Super-Natural Instructions

Super-Natural Instructions:
1.6K tasks, 3M+ examples

Classification, sequence tagging, rewriting/paraphrasing, translation, question answering..

Many (576+) languages!

Instruction Fine-Tuning — Example

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

[Chung et al., 2022]

Instruction Fine-Tuning — Example

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

PaLM 540B output

The reporter and the chef will discuss their favorite dishes.
The reporter and the chef will discuss the reporter's favorite dishes.
The reporter and the chef will discuss the chef's favorite dishes.
The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

✖ **(doesn't answer question)**

[Chung et al., 2022]

Instruction Fine-Tuning — Example

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

Flan-PaLM 540B output

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). ✅

[Chung et al., 2022]

Scaling Instruction Fine-Tuning

Model generation performance is positively correlated with **observed tasks and model size**

Number of examples does not have a big influence

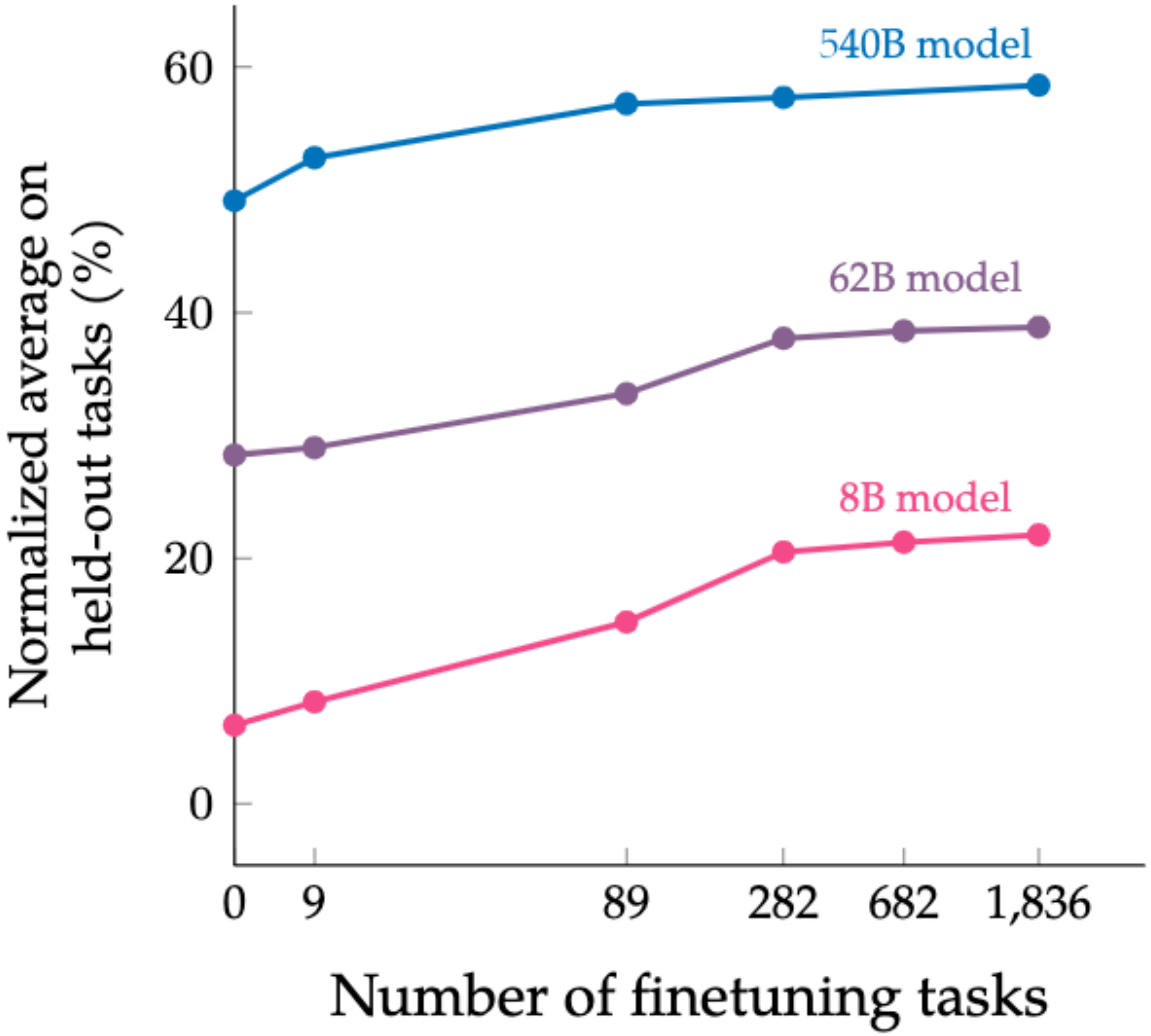
[Wang et al., 2022]

Scaling Instruction Fine-Tuning

Instruction Fine-Tuning improves the downstream performance on held-out tasks

Increasing the number of fine-tuning tasks improves generalisation

Increasing model scale by an order of magnitude (e.g., 8B → 62B, 62B → 540B) also help a lot



[Wang et al., 2022]

From LLMs to Assistants/Agents

Emergent Zero-Shot (ZS) and **Few-Shots** (FS) In-Context Learning Abilities:

- ✅ Can learn a task **without fine-tuning**
- ❌ Results are **highly sensitive** to the prompt being used
- ❌ Limited to what you can **fit in the input context**

Instruction Fine-Tuning

- ✅ **Simple** and **improves generalisation**
- ❌ Ground-truth data for tasks can be **expensive to collect**
- ❌ Open-ended generation tasks have **no single gold answer**

Reinforcement Learning with Human Feedback

Reward Model ~ Human Preferences

We are training a model on some task — e.g., to behave as a **personal assistant** for tasks like writing e-mails. For each sample s, assume we have a way to obtain a *human reward* for that sample: $R(s) \in \mathbb{R}$

Reward Model ~ Human Preferences

We are training a model on some task — e.g., to behave as a **personal assistant** for tasks like writing e-mails. For each sample s, assume we have a way to obtain a *human reward* for that sample: $R(s) \in \mathbb{R}$

```
Subject: Immediate Action
Required: Complete Your
Cybersecurity Training
Dear Team,
This is your final reminder to
complete the mandatory
cybersecurity training. Failure to
complete the training by the end
of this week will result [..]
```

$$R(s_1) = -2.5$$

Reward Model ~ Human Preferences

We are training a model on some task — e.g., to behave as a **personal assistant** for tasks like writing e-mails. For each sample s, assume we have a way to obtain a *human reward* for that sample: $R(s) \in \mathbb{R}$

```
Subject: Immediate Action
Required: Complete Your
Cybersecurity Training
Dear Team,
This is your final reminder to
complete the mandatory
cybersecurity training. Failure to
complete the training by the end
of this week will result [..]
```

$$R(s_1) = -2.5$$

```
Subject: Friendly Reminder:
Cybersecurity Training Deadline
Approaching
Hello Everyone,
Just a friendly reminder that the
deadline to complete our mandatory
cybersecurity training is fast
approaching. Please make sure to
complete it by the end of this week.
It's a great opportunity [..]
```

$$R(s_1) = 12.0$$

Reward Model ~ Human Preferences

We are training a model on some task — e.g., to behave as a **personal assistant** for tasks like writing e-mails. For each sample s, assume we have a way to obtain a *human reward* for that sample: $R(s) \in \mathbb{R}$

```
Subject: Immediate Action
Required: Complete Your
Cybersecurity Training
Dear Team,
This is your final reminder to
complete the mandatory
cybersecurity training. Failure to
complete the training by the end
of this week will result [..]
```

$$R(s_1) = -2.5$$

```
Subject: Friendly Reminder:
Cybersecurity Training Deadline
Approaching
Hello Everyone,
Just a friendly reminder that the
deadline to complete our mandatory
cybersecurity training is fast
approaching. Please make sure to
complete it by the end of this week.
It’s a great opportunity [..]
```

$$R(s_1) = 12.0$$

Now we want to maximise the **expected reward:**

$$\boxed{\mathbb{E}_{\hat{s} \sim p(s)} \left[R\left(\hat{s}\right) \right]}$$

Optimising for Human Preferences

Imagine a reward function $R(s)$ for any generation s

The reward is **higher** when humans **prefer** the generation

Improving the generation is equivalent to maximising the expected reward:

$$\mathbb{E}_{\boxed{\hat{s} \sim p_\theta(s)}} \left[\boxed{R(\hat{s})} \right]$$

Expected reward over the course of sampling from our model

$p_\theta(s)$ is a model with parameters θ we aim to optimise

Reward function encoding human preferences

Optimising for Human Preferences

Imagine we have a reward function $R(s)$ for any generation s

The reward is **higher** when humans **prefer** the generation

Improving the generation is equivalent to maximising the expected reward:

$$\mathbb{E}_{\hat{s} \sim p_\theta(s)} \left[R\left(\hat{s}\right) \right]$$

We want to:

Find the **best generative model** p_θ that maximises the expected reward:

$$\hat{\theta} = \arg\max_\theta \mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right]$$

Estimate the **reward function** encoding **human preferences** $R\left(s\right)$

Optimising the Generative Model p_θ

How do we change our model (LM) parameters θ to maximise this?

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_{\hat{s}\sim p_\theta}\left[R\left(\hat{s}\right)\right]$$

Optimising the Generative Model p_θ

How do we change our model (LM) parameters θ to maximise this?

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right]$$

We can use good old **gradient-based optimisation** (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]$$

But **how can we do that?**

Optimising the Generative Model p_θ

How do we change our model (LM) parameters θ to maximise this?

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right]$$

We can use good old **gradient-based optimisation** (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]$$

But **how can we do that?**

We can use **policy gradient methods**, e.g., REINFORCE [Williams, 1992], also referred to as the *score function estimator*, to estimate the gradient of the expected reward.

REINFORCE [Williams, 1992] 101

$$\boxed{\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]}$$

REINFORCE [Williams, 1992] 101

$$\boxed{\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]} = \nabla_\theta \left[\sum_s p_\theta(s) R(s) \right]$$

Def. of expectation

REINFORCE [Williams, 1992] 101

$$\boxed{\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]} = \nabla_\theta \left[\sum_s p_\theta(s) R(s) \right]$$

Def. of expectation

$$= \sum_s R(s) \nabla_\theta p_\theta(s)$$

Gradient distributes over the sum

REINFORCE [Williams, 1992] 101

$$\boxed{\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]} = \nabla_\theta \left[\sum_s p_\theta(s) R(s) \right]$$

Def. of expectation

$$= \sum_s R(s) \boxed{\nabla_\theta p_\theta(s)}$$

Gradient distributes over the sum

$$= \sum_s R(s) \boxed{p_\theta(s) \nabla_\theta \log p_\theta(s)}$$

$$\nabla_\theta \log p_\theta(s) = \frac{\nabla_\theta p_\theta(s)}{p_\theta(s)}$$

REINFORCE [Williams, 1992] 101

$$\boxed{\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]} = \nabla_\theta \left[\sum_s p_\theta(s) R(s) \right]$$

Def. of expectation

$$= \sum_s R(s) \boxed{\nabla_\theta p_\theta(s)}$$

Gradient distributes over the sum

$$= \sum_s R(s) \boxed{p_\theta(s) \nabla_\theta \log p_\theta(s)}$$

$$\nabla_\theta \log p_\theta(s) = \frac{\nabla_\theta p_\theta(s)}{p_\theta(s)}$$

Def. of expectation

$$= \mathbb{E}_{p_\theta} \left[R(s) \nabla_\theta \log p_\theta(s) \right]$$

REINFORCE [Williams, 1992] 101

$$\boxed{\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]} = \nabla_\theta \left[\sum_s p_\theta(s) R(s) \right]$$

Def. of expectation

$$= \sum_s R(s) \boxed{\nabla_\theta p_\theta(s)}$$

Gradient distributes over the sum

$$= \sum_s R(s) \boxed{p_\theta(s) \nabla_\theta \log p_\theta(s)}$$

$$\nabla_\theta \log p_\theta(s) = \frac{\nabla_\theta p_\theta(s)}{p_\theta(s)}$$

Def. of expectation

$$= \mathbb{E}_{p_\theta} \left[R(s) \nabla_\theta \log p_\theta(s) \right]$$

$$\boxed{\approx \frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_\theta \log p_\theta(s_i)}$$ with $s_i \sim p_\theta$

Monte Carlo estimate

REINFORCE [Williams, 1992] 101

$$\boxed{\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_{\theta_t}} \left[R\left(\hat{s}\right) \right] \right]} = \nabla_\theta \left[\sum_s p_\theta(s) R(s) \right]$$

Def. of expectation

$$= \sum_s R(s) \boxed{\nabla_\theta p_\theta(s)}$$

Gradient distributes over the sum

$$= \sum_s R(s) \boxed{p_\theta(s) \nabla_\theta \log p_\theta(s)}$$

$$\nabla_\theta \log p_\theta(s) = \frac{\nabla_\theta p_\theta(s)}{p_\theta(s)}$$

Def. of expectation

$$= \mathbb{E}_{p_\theta} \left[R(s) \nabla_\theta \log p_\theta(s) \right]$$

$$\boxed{\approx \frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_\theta \log p_\theta(s_i)} \text{ with } s_i \sim p_\theta$$

Monte Carlo estimate

Optimising the Generative Model p_θ

How do we change our model (LM) parameters θ to maximise this?

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right]$$

We can use good old **gradient-based optimisation** (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \boxed{\left[\frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_\theta \log p_\theta(s_i) \right]} \quad \text{with } s_i \sim p_\theta$$

REINFORCE estimate of

$$\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right] \right]$$

Optimising the Generative Model p_θ

How do we change our model (LM) parameters θ to maximise this?

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right]$$

We can use good old **gradient-based optimisation** (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \boxed{\left[\frac{1}{n} \sum_{i=1}^{n} R(s_i) \nabla_\theta \log p_\theta(s_i) \right]} \quad \text{with } s_i \sim p_\theta$$

REINFORCE estimate of $\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right] \right]$

Exercise — what if $R(s) \in \{0,1\}$**?** 🙂

Optimising the Generative Model p_θ

How do we change our model (LM) parameters θ to maximise this?

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right]$$

We can use good old **gradient-based optimisation** (gradient ascent):

$$\theta_{t+1} \leftarrow \theta_t + \alpha \boxed{\left[\frac{1}{n} \sum_{i=1}^{n} R(s_i) \, \nabla_\theta \log p_\theta(s_i) \right]} \quad \text{with } s_i \sim p_\theta$$

REINFORCE estimate of
$$\nabla_\theta \left[\mathbb{E}_{\hat{s} \sim p_\theta} \left[R\left(\hat{s}\right) \right] \right]$$

Exercise — what if $R(s) \in \{0,1\}$? 🙂

Note: this was heavily simplified — in reality, it can require many tricks to work 😬

How do we optimise for human preferences?

Recap — given an **arbitrary**, **non differentiable reward function** $R(s)$, we can train our LM to maximise the expected reward! However —

How do we optimise for human preferences?

Recap — given an **arbitrary**, **non differentiable reward function** $R(s)$, we can train our LM to maximise the expected reward! However —

1. **Having a human in the loop to assign reward values is costly!**

How do we optimise for human preferences?

Recap — given an **arbitrary**, **non differentiable reward function** $R(s)$, we can train our LM to maximise the expected reward! However —

1. **Having a human in the loop to assign reward values is costly!**

   Solution: collect some human preferences, and train another model (the **reward model**) to predict new human preferences [Knox et al., 2009]

How do we optimise for human preferences?

Recap — given an **arbitrary**, **non differentiable reward function** $R(s)$, we can train our LM to maximise the expected reward! However —

1. Having a human in the loop to assign reward values is costly!

Solution: collect some human preferences, and train another model (the **reward model**) to predict new human preferences [Knox et al., 2009]

2. Human judgements tend to be noisy/mis-calibrated!

How do we optimise for human preferences?

Recap — given an **arbitrary**, **non differentiable reward function** $R(s)$, we can train our LM to maximise the expected reward! However —

1. **Having a human in the loop to assign reward values is costly!**

   Solution: collect some human preferences, and train another model (the **reward model**) to predict new human preferences [Knox et al., 2009]

2. **Human judgements tend to be noisy/mis-calibrated!**

   Solution: rather than asking for direct ratings, ask for **pairwise comparisons**, which tend to be more reliable [Clark et al., 2018]

RLHF — Putting it all together

For Reinforcement Learning with Human Feedback, we have:

- A **pre-trained** — possibly **instruction fine-tuned** — **LM** $p^{\text{LM}}(s)$
- A **reward model** $\text{RM}(s)$

To optimise our model, we:

- Create a copy of the model $p_\theta^{\text{RL}}(s)$ with parameters θ
- Optimise for the following reward with RL:

$$R(s) = \text{RM}(s) - \beta \log \boxed{\left[\frac{p_\theta^{\text{RL}}(s)}{p^{\text{LM}}(s)}\right]}$$

Pay a price when $p_\theta^{\text{RL}}(s) > p^{\text{LM}}(s)$

[Stiennon et al. 2020]

Reading List

Improving Language Understanding by Generative Pre-Training, **https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf**

Language Models are Unsupervised Multitask Learners, **https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf**

The LAMBADA dataset: Word prediction requiring a broad discourse context, **https://aclanthology.org/P16-1144/**