



COMP 336 I Natural Language Processing

Lecture 12: LLM prompting, in-context learning,
scaling laws, emergent capacities

Spring 2024

Announcements

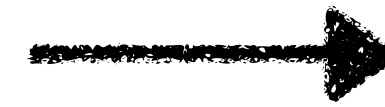
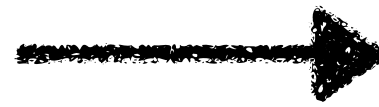
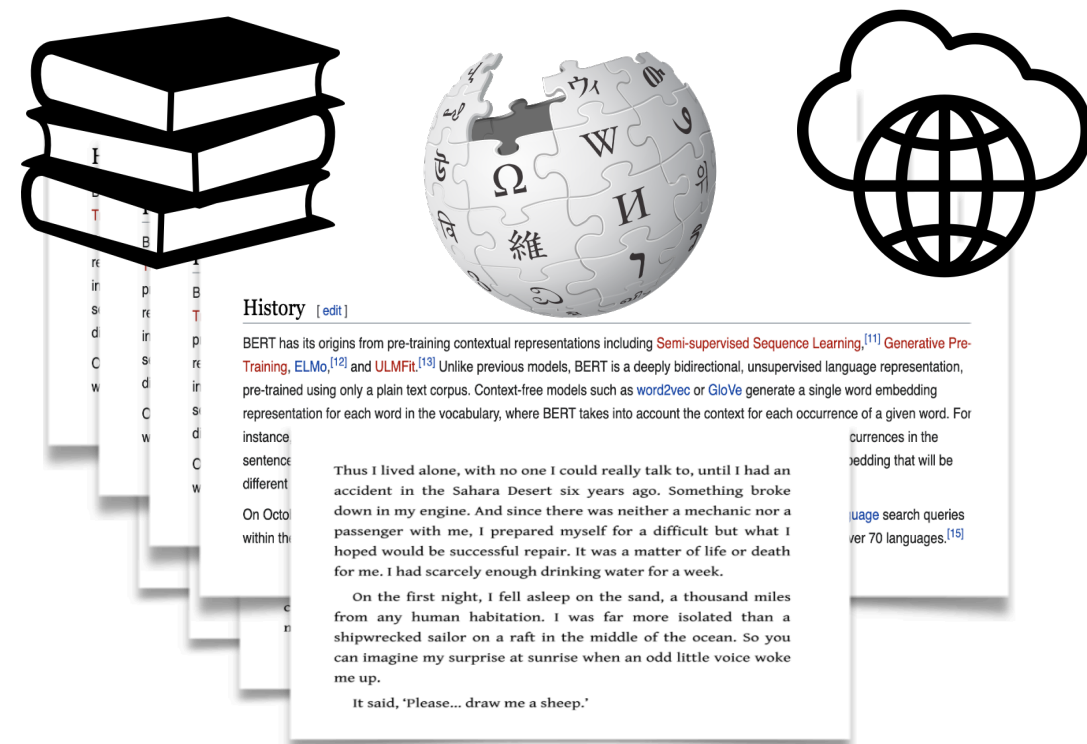
- Final exam is scheduled at 9:30 - 11:30am on May 8, Wed @Rm 3 Library Ext.
- [#assignment-2](#) due next week!
 - Join [#assignment-2](#) Slack channel for discussion

Lecture plan

- LLM pretraining objectives: recap
- LLM prompting and in-context learning
- Scaling laws of LLMs
- Emergent capacities of LLMs

Pretraining: training objectives?

- During pretraining, we have a large text corpus (**no task labels**)
 - **Key question: what labels or objectives used to train the vanilla Transformers?**



**Training
labels/objectives?**

Pretraining Transformers

Pretraining objectives



BERT (Encoder-only)

Devlin et al., 2018

The cabs ___ the same rates as those ___ by horse-drawn cabs and were ___ quite popular, ___ the Prince of Wales (the ___ King Edward VII) travelled in ___. The cabs quickly ___ known as "hummingbirds" for ___ noise made by their motors and their distinctive black and ___ livery. Passengers ___ the interior fittings were ___ when compared to ___ cabs but there ___ some complaints ___ the ___ lighting made them too ___ to those outside ___.

charged, used, initially, even, future, became, the, yellow, reported, that, luxurious, horse-drawn, were that, internal, conspicuous, cab

Masked token prediction



T5 (Encoder-decoder)

Raffel et al., 2019

Original text

Thank you ~~for inviting~~ me to your party last week.

Inputs

Thank you <X> me to your party <Y> week.

Targets

<X> for inviting <Y> last <Z>

Denosing span-mask prediction



GPT - 4

Decoder-only

Text: Second Law of Robotics: A robot must obey the orders given it by human beings

Generated training examples

Example #	Input (features)	Correct output (labels)
1	Second law of robotics :	a
2	Second law of robotics : a	robot
3	Second law of robotics : a robot	must
...		

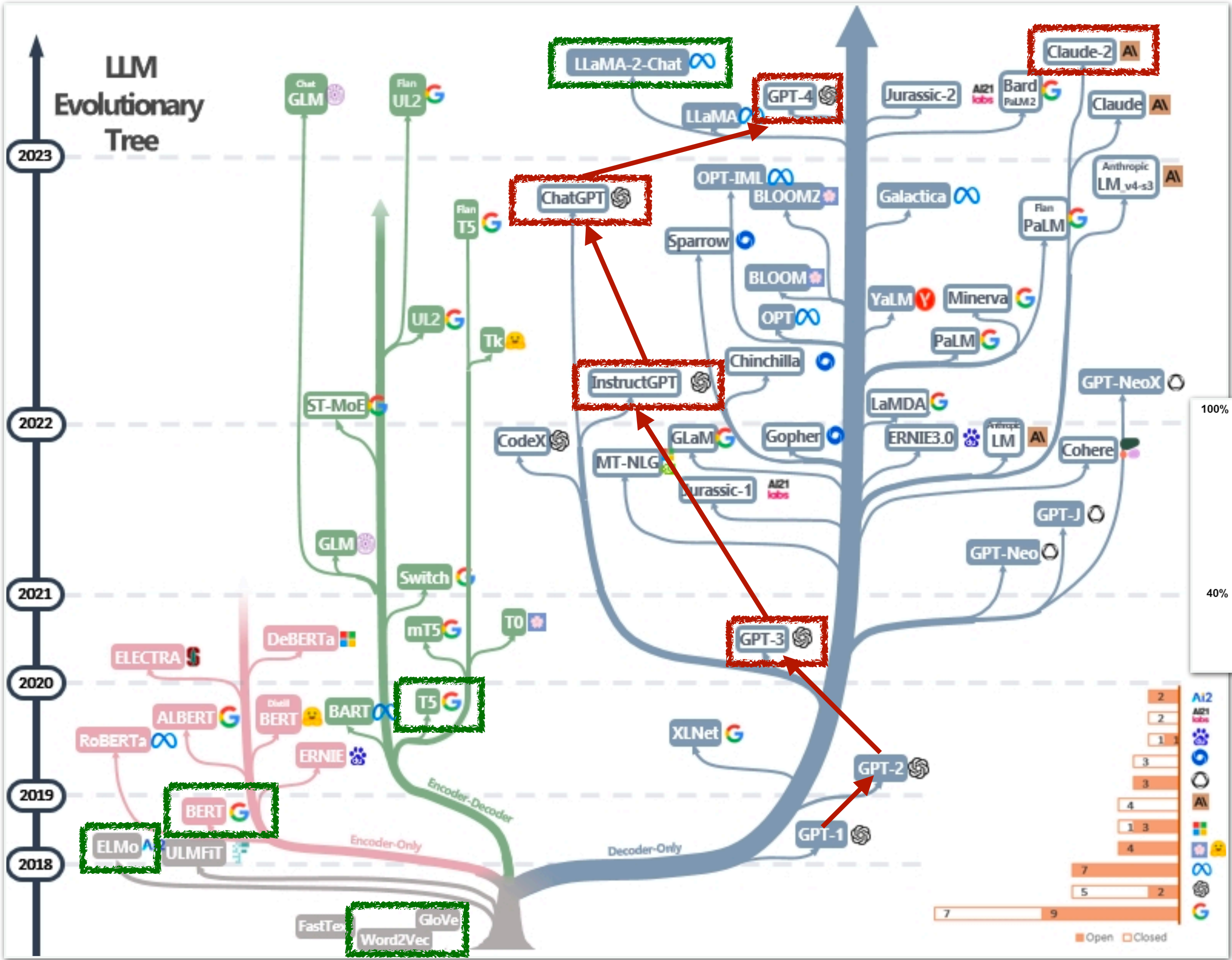
Next token prediction

Evolution tree of pretrained LMs

~200 billion

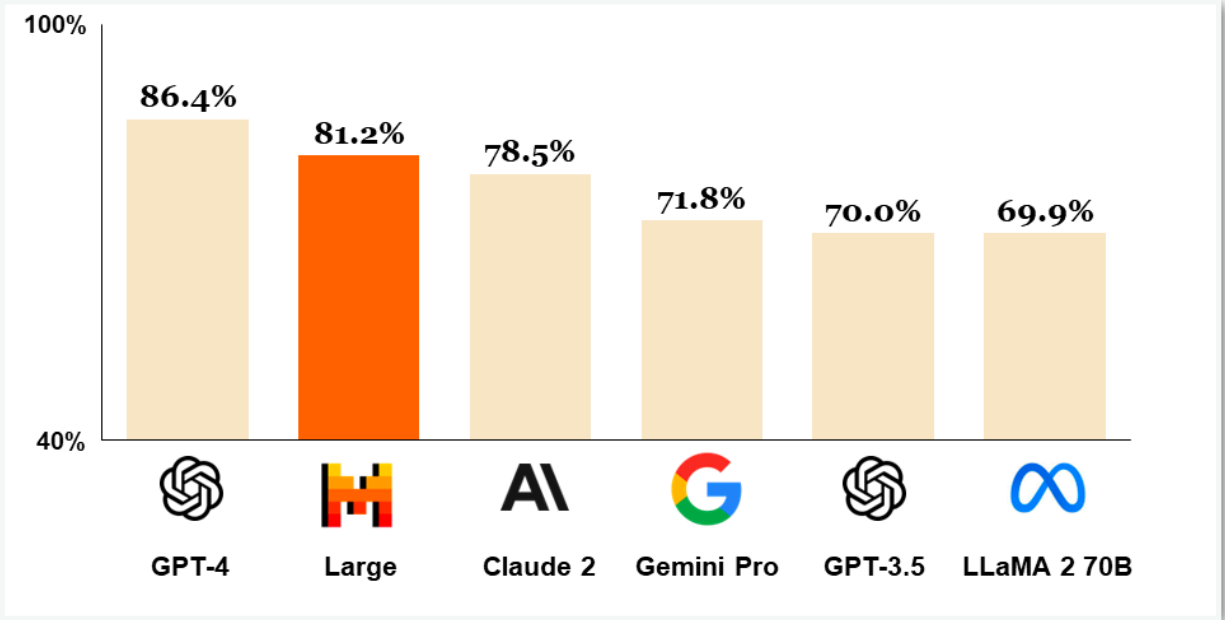
Model size
(# of parameters)
~1000 times larger

~300 million



Open-sourced

Close-sourced



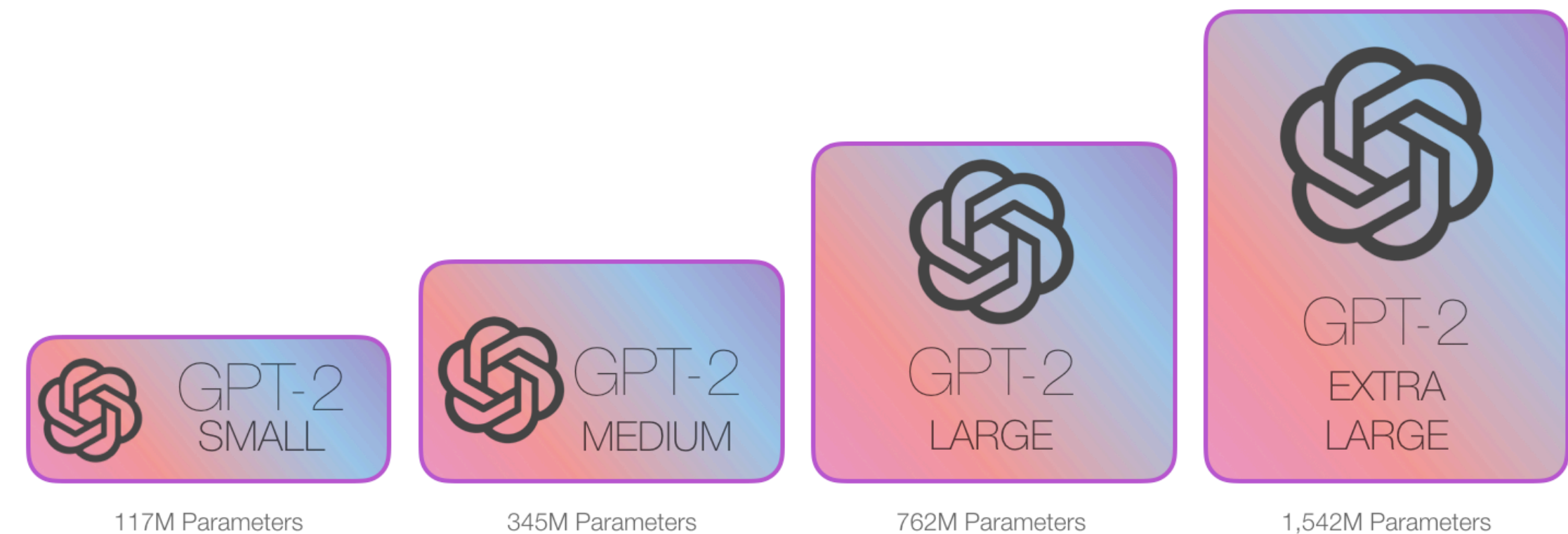
From GPT-1 to GPT-2 to GPT-3

- All decoder-only Transformer-based language models
- Model size ↑, training corpora ↑

GPT-2



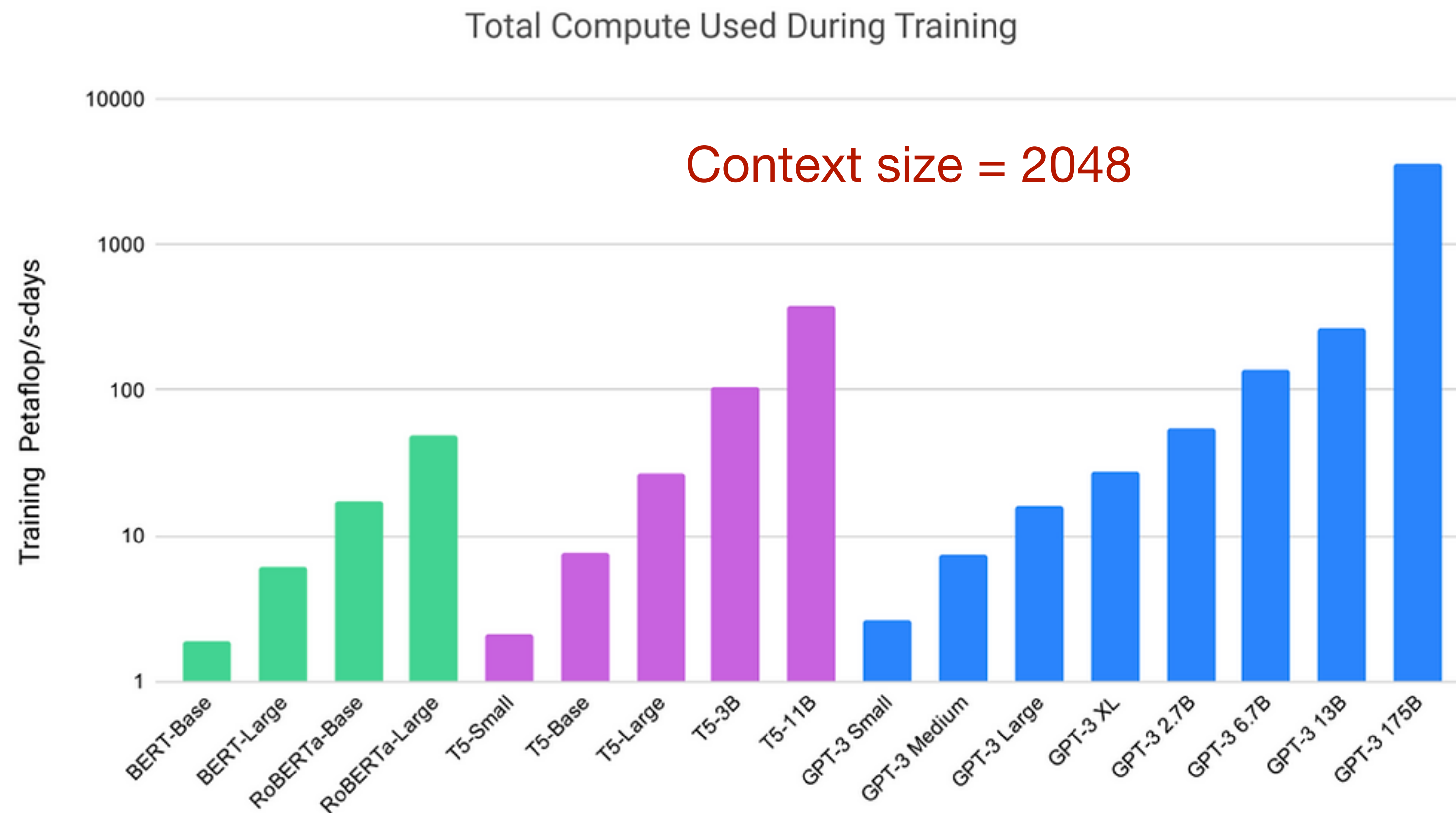
Context size = 1024



.. trained on 40Gb of Internet text ..

GPT-3: language models are few-shot learners

- GPT-2 → GPT-3: 1.5B → **175B** (# of parameters), ~14B → **300B** (# of tokens)



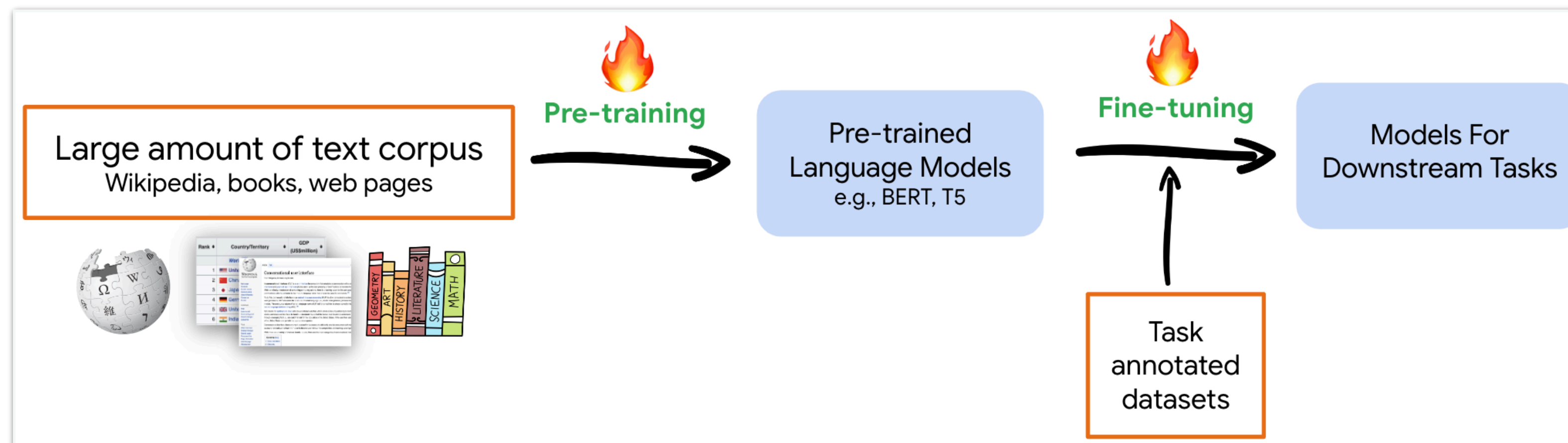
Training computation is measured using floating-point operations or “FLOP”.

One FLOP represents a single arithmetic operation involving floating-point numbers, such as addition, subtraction, multiplication, or division.

Before GPT3: Modern learning paradigm

- **Pre-training + supervised training/fine-tuning**

- First train Transformer using a lot of general text using unsupervised learning. This is called **pretraining**.
- Then train the pretrained Transformer for a specific task using supervised learning. This is called **finetuning**.



Paradigm shift since GPT-3

- Before GPT-3, **Pre-training + supervised training/fine-tuning** is the default way of doing learning in models like BERT/T5/GPT-2
 - SST-2 has 67k examples, SQuAD has 88k (passage, answer, question) triples
- Fine-tuning requires computing the gradient and applying a parameter update on every example (or every K examples in a mini-batch)
- However, this is very expensive for the 175B GPT-3 model

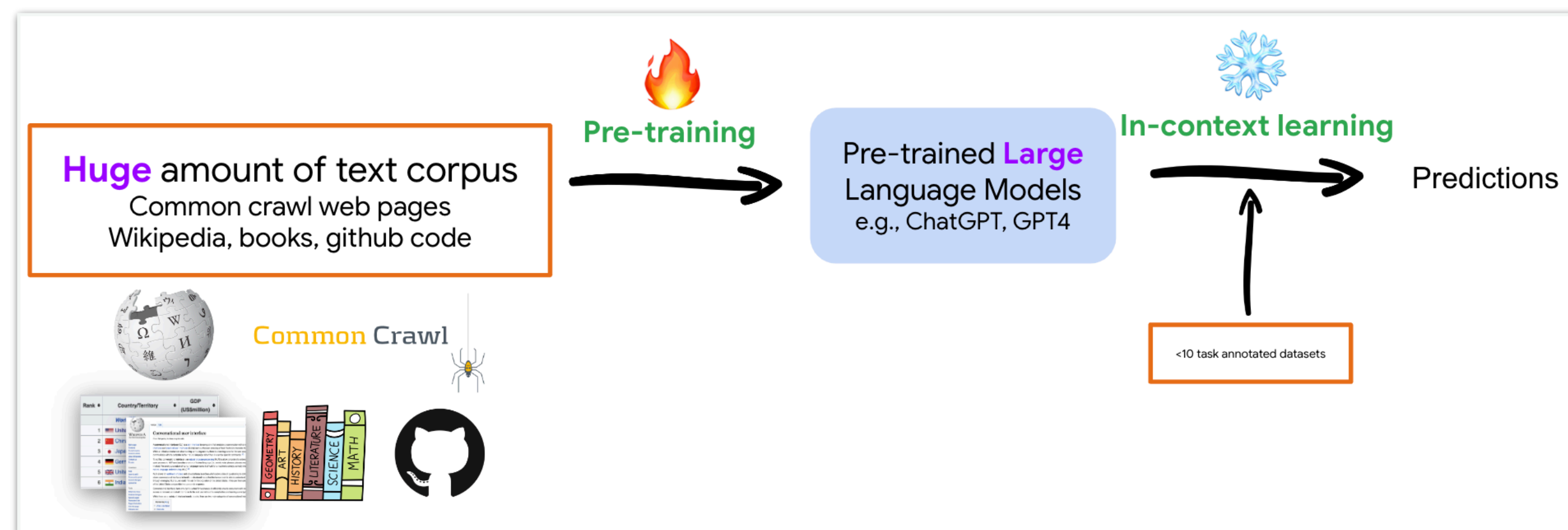
Fine-tuning

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



Latest learning paradigm shift since GPT-3

- **Pre-training + prompting/in-context learning (no training this step)**
 - First train a **large (>7~175B)** Transformer using a lot of general text using unsupervised learning. This is called **large** language model **pretraining**.
 - Then **directly use** the pretrained large Transformer (**no further finetuning/training**) for any different task given only a natural language description of the task or a few task (x, y) examples. This is called **prompting/in-context learning**.

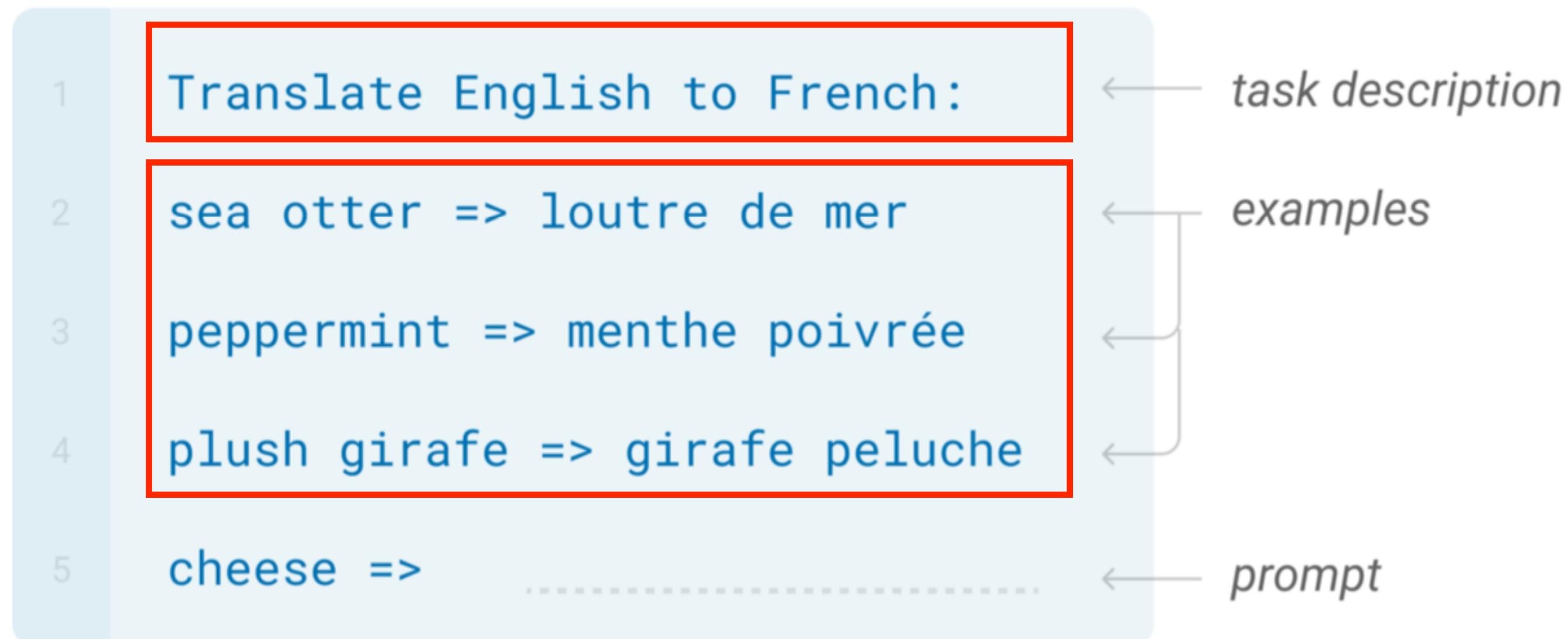


GPT-3: few-shot in-context learning

- GPT-3 proposes an alternative: **in-context learning**

Few-shot

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



- This is just a forward pass, **no gradient update at all!**
- You only need to feed a small number of examples (e.g., 32)
(On the other hand, you can't feed many examples at once too as it is bounded by context size)

GPT-3: task specifications

Context → Passage: Saint Jean de Brébeuf was a French Jesuit missionary who travelled to New France in 1625. There he worked primarily with the Huron for the rest of his life, except for a few years in France from 1629 to 1633. He learned their language and culture, writing extensively about each to aid other missionaries. In 1649, Brébeuf and another missionary were captured when an Iroquois raid took over a Huron village . Together with Huron captives, the missionaries were ritually tortured and killed on March 16, 1649. Brébeuf was beatified in 1925 and among eight Jesuit missionaries canonized as saints in the Roman Catholic Church in 1930.
Question: How many years did Saint Jean de Brébeuf stay in New France before he went back to France for a few years?
Answer:

Target Completion → 4

DROP
(a reading comprehension task)

Context → Please unscramble the letters into a word, and write that word:
skicts =

Target Completion → sticks

Unscrambling words

Context → An outfitter provided everything needed for the safari.
Before his first walking holiday, he went to a specialist outfitter to buy some boots.
question: Is the word 'outfitter' used in the same way in the two sentences above?
answer:

Target Completion → no

Word in context (WiC)

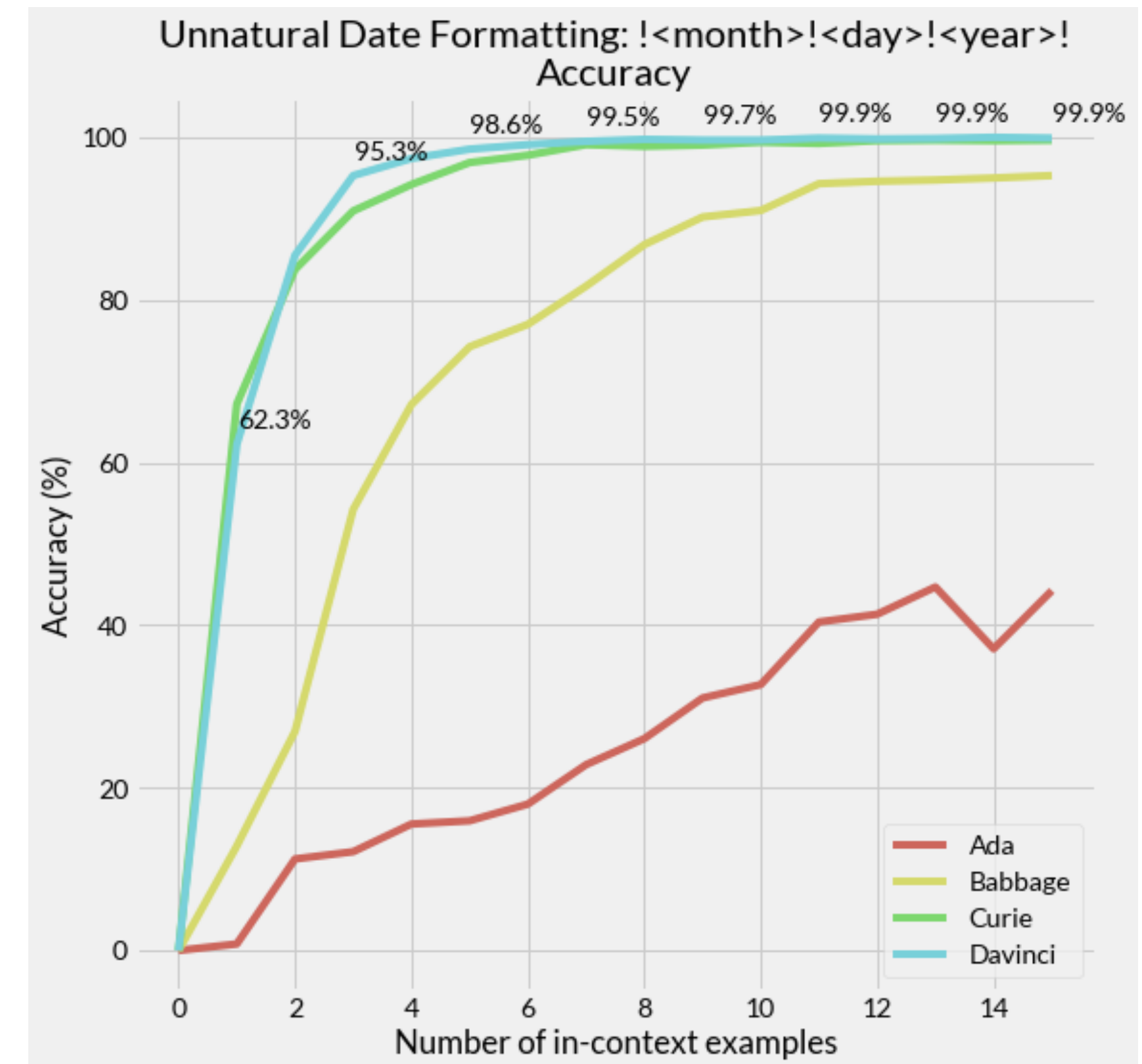
GPT-3's in-context learning

Input: 2014-06-01
Output: !06!01!2014!
Input: 2007-12-13
Output: !12!13!2007!
Input: 2010-09-23
Output: !09!23!2010!
Input: **2005-07-23**
Output: **!07!23!2005!**

in-context examples

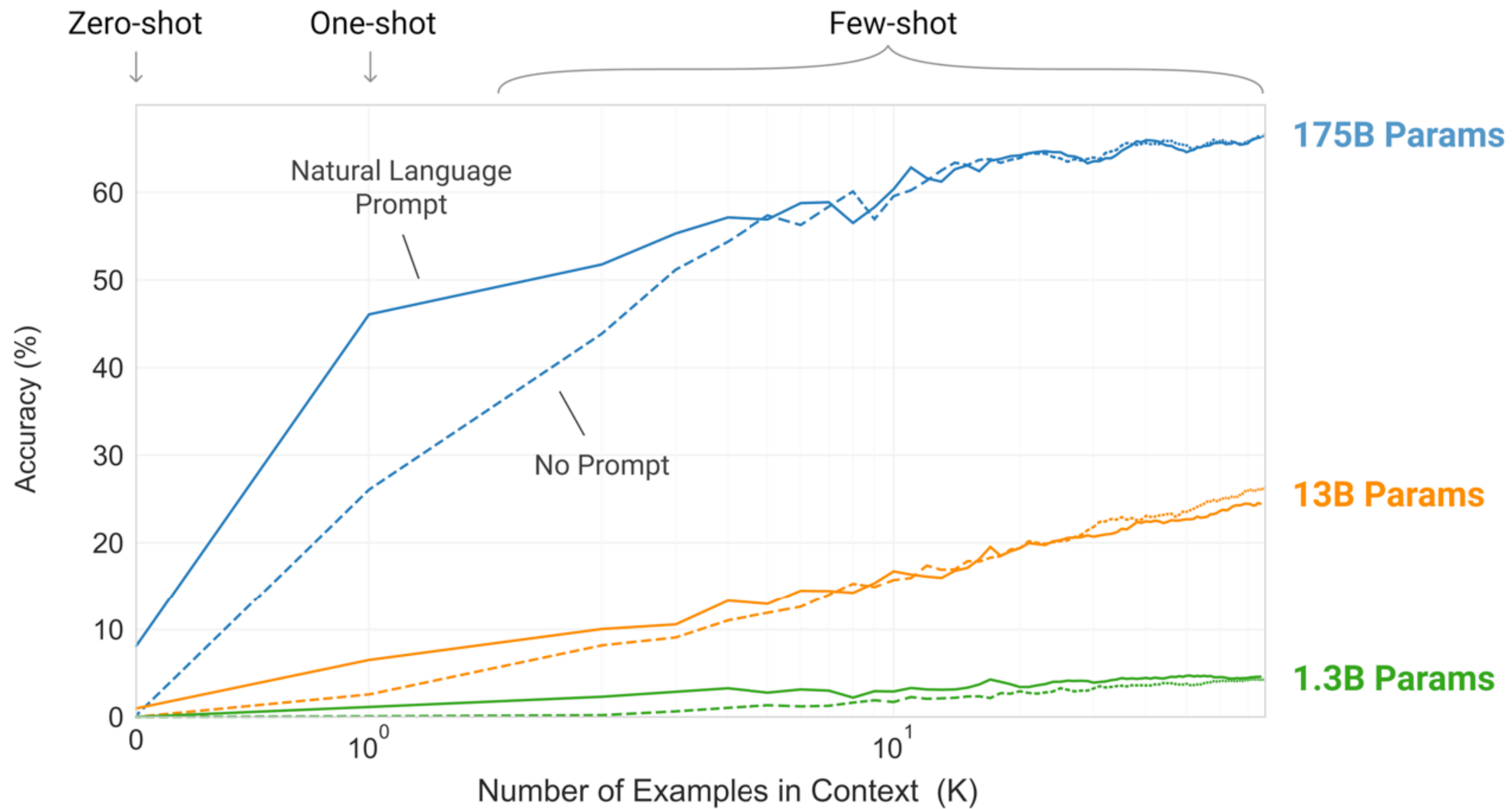
test example

model completion



<http://ai.stanford.edu/blog/in-context-learning/>

GPT-3's scaling laws in performance



Chain-of-thought (CoT) prompting

Standard Prompting

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain of Thought Prompting

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Why in-context learning with LLMs?

- Amazing zero/few-shot performance
 - Save a lot of annotation! 🎉
- Easy to use without training
 - Just talk to them! 👍
- One model for many NLP applications 😊
 - No need to annotate and fine-tune for different tasks

But, again, they are sensitive to prompts! Need to design a good prompt or train a good example retriever! 😂

Okay, so bigger is better? Can you be more specific?

Scaling Laws

Scaling Laws (Kaplan et al., 2020)

- Kaplan et al., 2020 (OpenAI) explore how performance scales w.r.t. several parameters
- Vary:
 - Scale: N - # Model Params, D - Dataset size (tokens)
 - Other hyperparameters: Hidden layer sizes, context length, batch size
- Goal: Can we reliably predict test loss L based on training scale (parameters and dataset size)?

Scaling Laws (Kaplan et al., 2020)

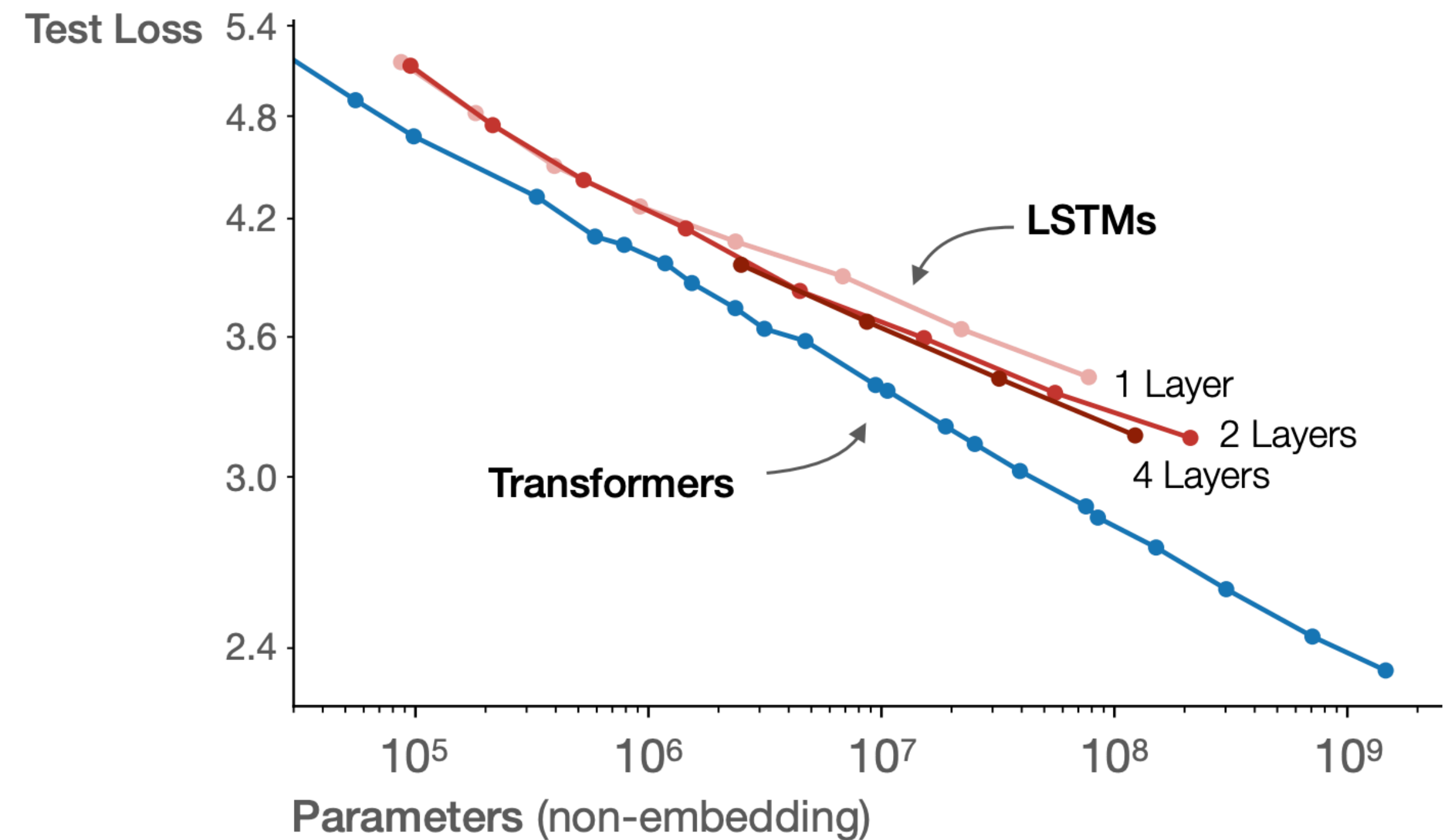
- Result: Test loss L very closely follows a *power law*:
- Given constant dataset size D ,

$$L(N) \approx \left(\frac{N_c}{N} \right)^{\alpha N}$$

- Given constant model size N ,

$$L(D) \approx \left(\frac{D_c}{D} \right)^{\alpha D}$$

To linearly decrease test loss L , you need to exponentially increase dataset size D or model size N



Scaling Laws (Kaplan et al., 2020)

- Result: Test loss L very closely follows a *power law*:

$$L(N) \approx \left(\frac{N_c}{N} \right)^{\alpha N}$$

- Given constant dataset size D ,

- Bringing it together:

$$L(N, D) \approx \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha N}{\alpha D}} + \frac{D_c}{D} \right]^{\alpha D}$$

- Given constant model size N ,

$$L(D) \approx \left(\frac{D_c}{D} \right)^{\alpha D}$$

Parameter	α_N	α_D	N_c	D_c
Value	0.076	0.103	6.4×10^{13}	1.8×10^{13}

← Empirical estimates of parameters from experiments

Table 2 Fits to $L(N, D)$

LLaMA (Touvron et al., 2023)

- OpenAI/Deepmind only looked at the optimal size given a fixed *training* compute budget

$$\operatorname{argmin}_{N, D \text{ s.t. } \text{FLOPs}(N, D) = C} L(N, D)$$

- What if you care more about *inference* time compute cost?
- Smaller model => Smaller inference cost
- To get best small model, should just train a small model on as much data as possible (beyond “Chinchilla-optimal”)
- “Overtrained” LLaMA-13B outperformed GPT-3 on many benchmarks

Recently

- A lot of recent progress has been made from training bigger models on more data: LLaMA 2, GPT-4, Gemini, Mistral, etc.
 - Note: quality matters too! Need more *high-quality data*, low-quality data does not improve performance
- Limits of scale:
 - Limits on data: Modern LLMs are trained on basically the *entire internet* - we can't find 10 new internets out of nowhere
 - Limits on compute: Big tech companies can't continue to 10x their model sizes for much longer

TECHNOLOGY | ARTIFICIAL INTELLIGENCE

Sam Altman Seeks Trillions of Dollars to Reshape Business of Chips and AI

OpenAI chief pursues investors including the U.A.E. for a project possibly requiring up to \$7 trillion

By [Keach Hagey](#) [Follow](#) and [Asa Fitch](#) [Follow](#)

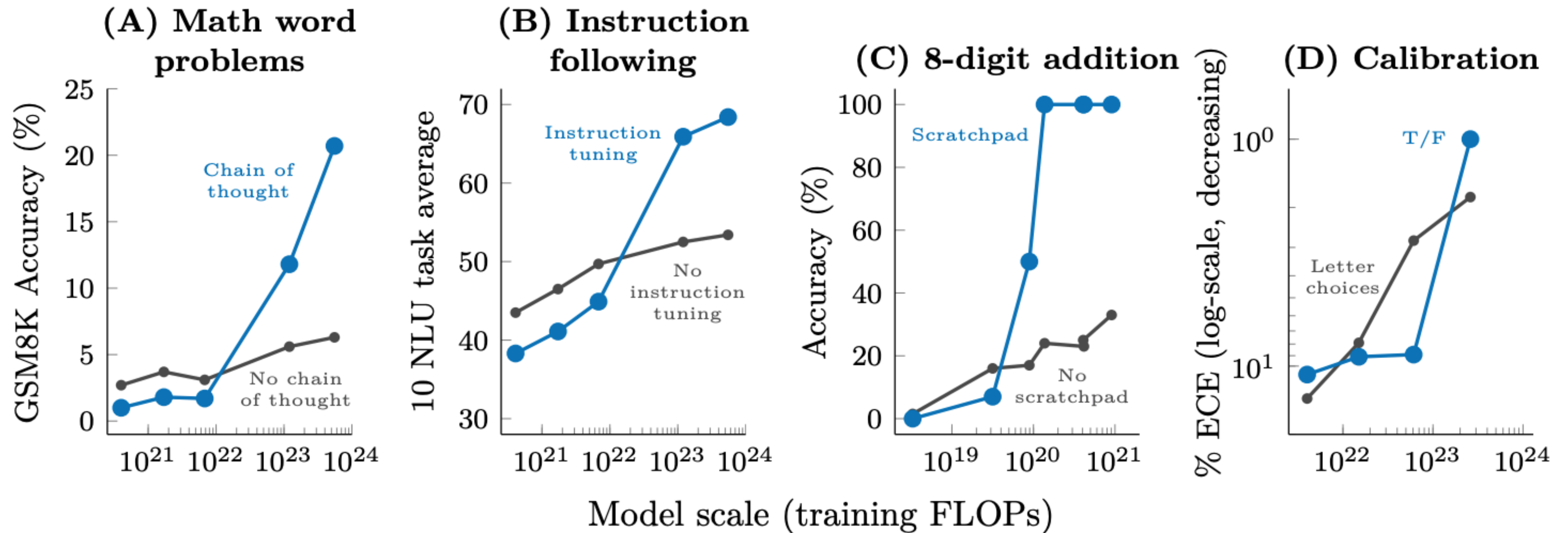
Feb. 8, 2024 9:00 pm ET

(For context: \$7T is more than GDP of all countries except US and China! Japan: \$4.2T, Germany: \$4T, ...)

But that won't stop Sam Altman from trying!

Emergent capabilities of LLMs?

Emergent properties of LLMs



(Wei et al., 2022) Emergent Abilities of Large Language Models

Emergent capabilities a mirage?

- ([Schaeffer et al., 2023](#)) take issue with the characterization of “emergent capabilities”
- Most metrics used in ([Wei et al., 2022](#)) were “hard” metrics which don’t give partial credit like accuracy

Hard Accuracy:

A) $123 + 456 = 579$ ✓

B) $123 + 456 = 578$ ✗

C) $123 + 456 = 42$ ✗

In ([Wei et al., 2022](#)), B and C are both wrong, even though B is much closer to correct than C

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo


Computer Science, Stanford University

Emergent capabilities a mirage?

- (Schaeffer et al., 2023) measure soft metrics (e.g., how many digits are correct, probability of the right answer) for “emergent abilities”
- Find much more predictable scaling
- Different metric choices lead to different appearances of “emergent” or not emergent
- “Emergent abilities” are a mirage(?)


Hard Accuracy:


A) $123 + 456 = 579$ 

B) $123 + 456 = 578$ 

C) $123 + 456 = 42$ 

Soft Accuracy (# correct digits):

A) $123 + 456 = 579$ 3/3 

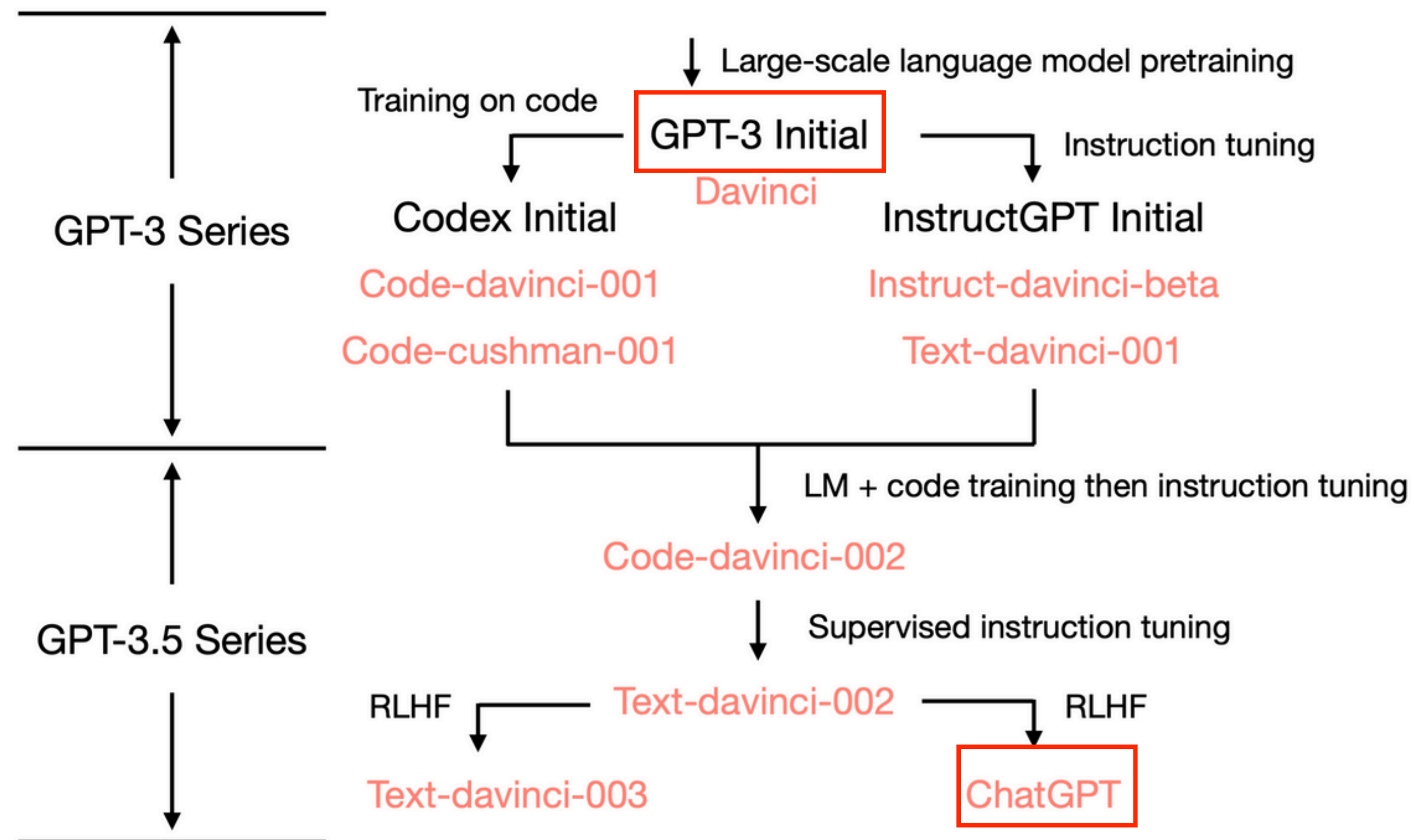
B) $123 + 456 = 578$ 2/3 

C) $123 + 456 = 42$ 0/3 

What happened after GPT-3?

(Is model size ↑, training corpora ↑ the only way to go?)

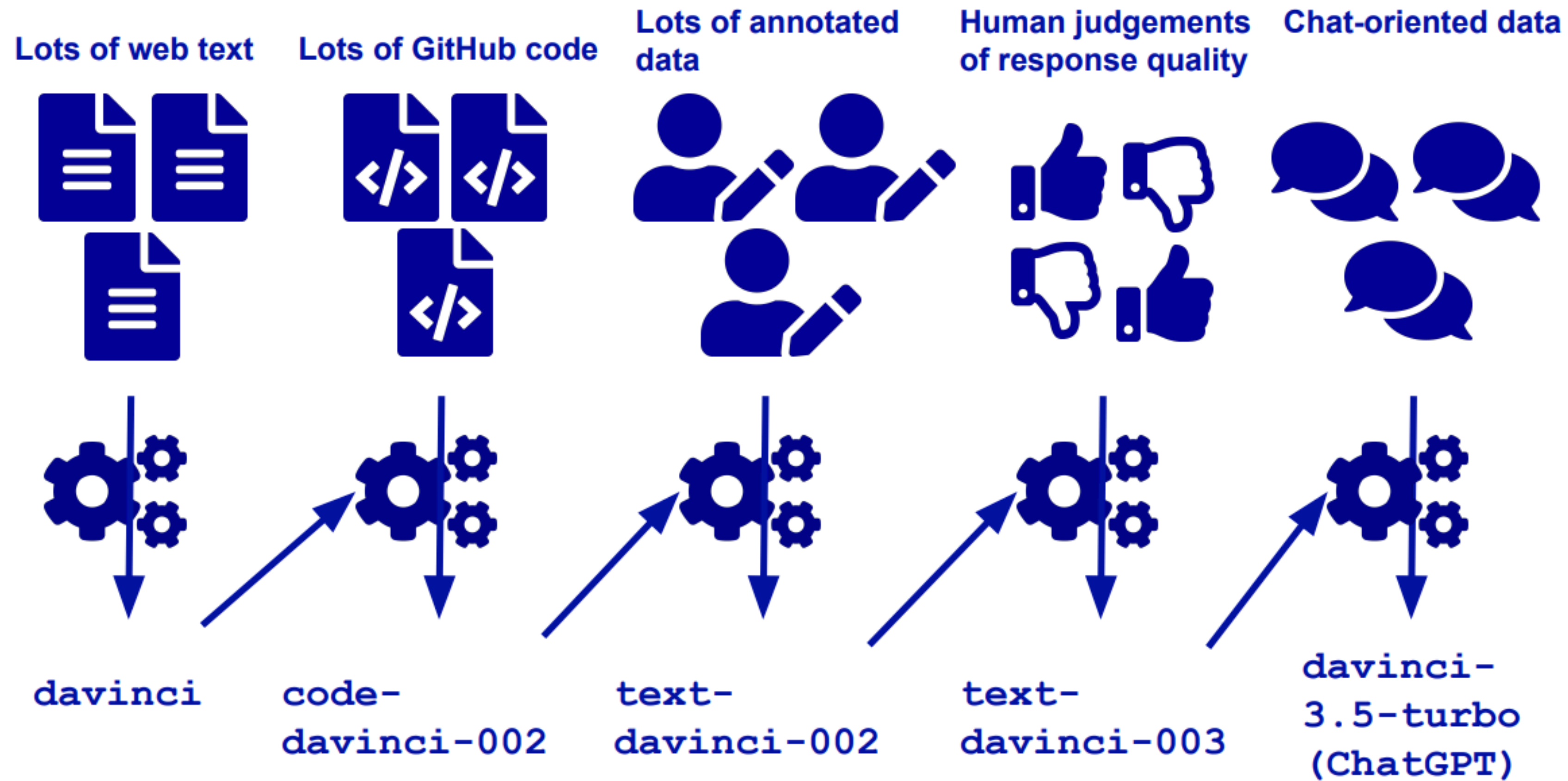
How was ChatGPT developed?



What's new?

- Training on code
- **Supervised instruction tuning**
- **RLHF = Reinforcement learning from human feedback**

How was ChatGPT developed?



(Slide credit: Graham Neubig)

InstructGPT: Supervised instruction tuning + RLHF

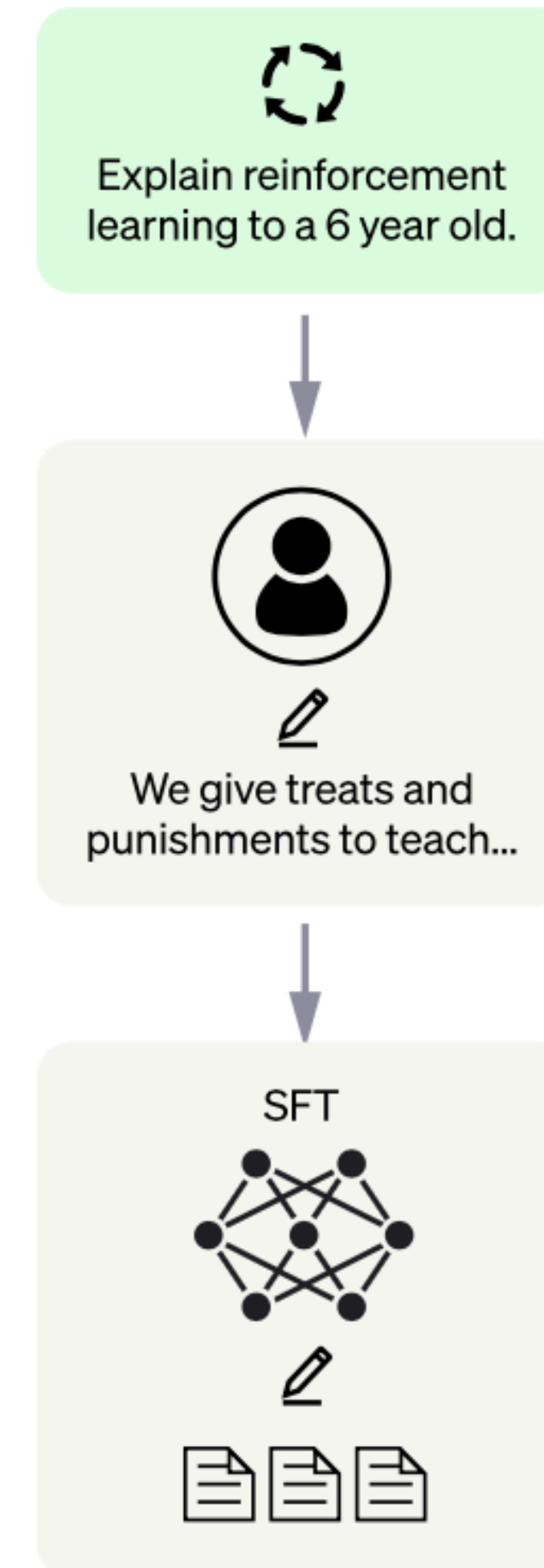
Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.



Supervised instruction tuning

Use-case	Prompt	Use-case	(%)	Number of Prompts		
Brainstorming	List five ideas for how to regain enthusiasm for my career	Generation	45.6%	SFT Data		
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.	Open QA	12.4%	split	source	size
Rewrite	This is the summary of a Broadway play: "" {summary} "" This is the outline of the commercial for that play: ""	Brainstorming	11.2%	train	labeler	11,295
		Chat	8.4%	train	customer	1,430
		Rewrite	6.6%	valid	labeler	1,550
		Summarization	4.2%	valid	customer	103
		Classification	3.5%			
		Other	3.5%			
		Closed QA	2.6%			
		Extract	1.9%			

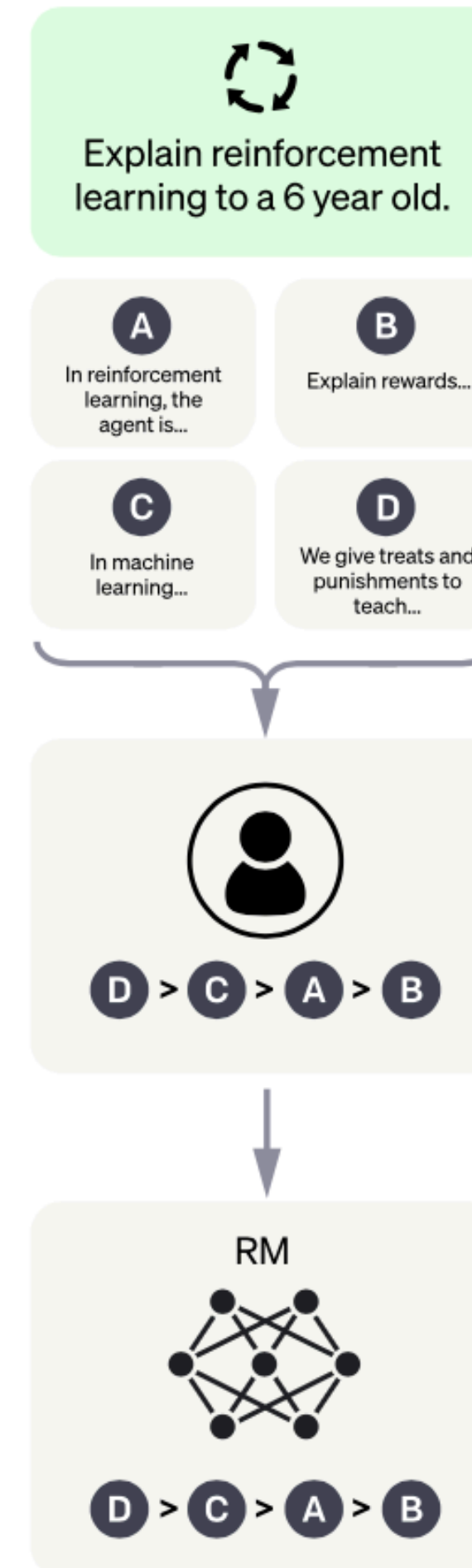
SFT data: only ~13k (not public)

InstructGPT: Supervised instruction tuning + RLHF

Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

InstructGPT: Supervised instruction tuning + RLHF

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

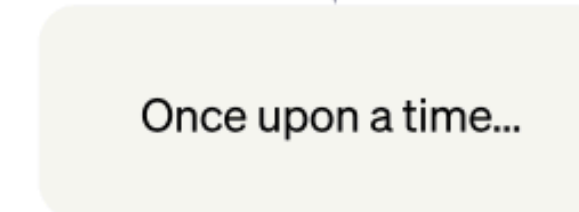
A new prompt is sampled from the dataset.



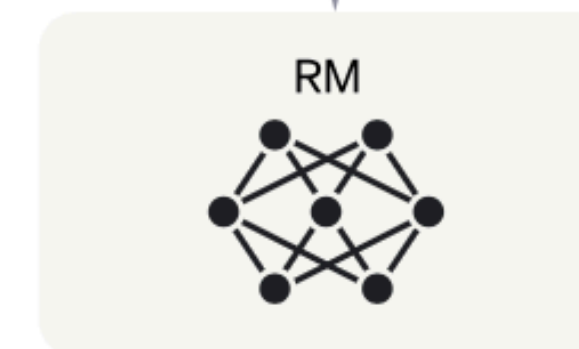
The PPO model is initialized from the supervised policy.



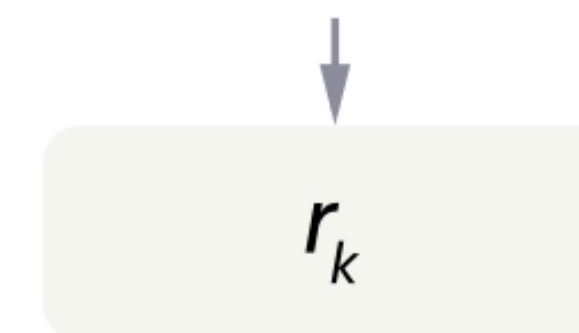
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



ChatGPT = InstructGPT + dialogue data

Introducing ChatGPT

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

“We trained this model using Reinforcement Learning from Human Feedback (RLHF), **using the same methods as InstructGPT**, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. **We mixed this new dialogue dataset with the InstructGPT dataset**, which we transformed into a dialogue format.”

Human feedback data is the key!

<https://openai.com/blog/chatgpt>

Recent models are getting smaller?

RESEARCH

Introducing LLaMA: A foundational, 65-billion-parameter large language model

February 24, 2023



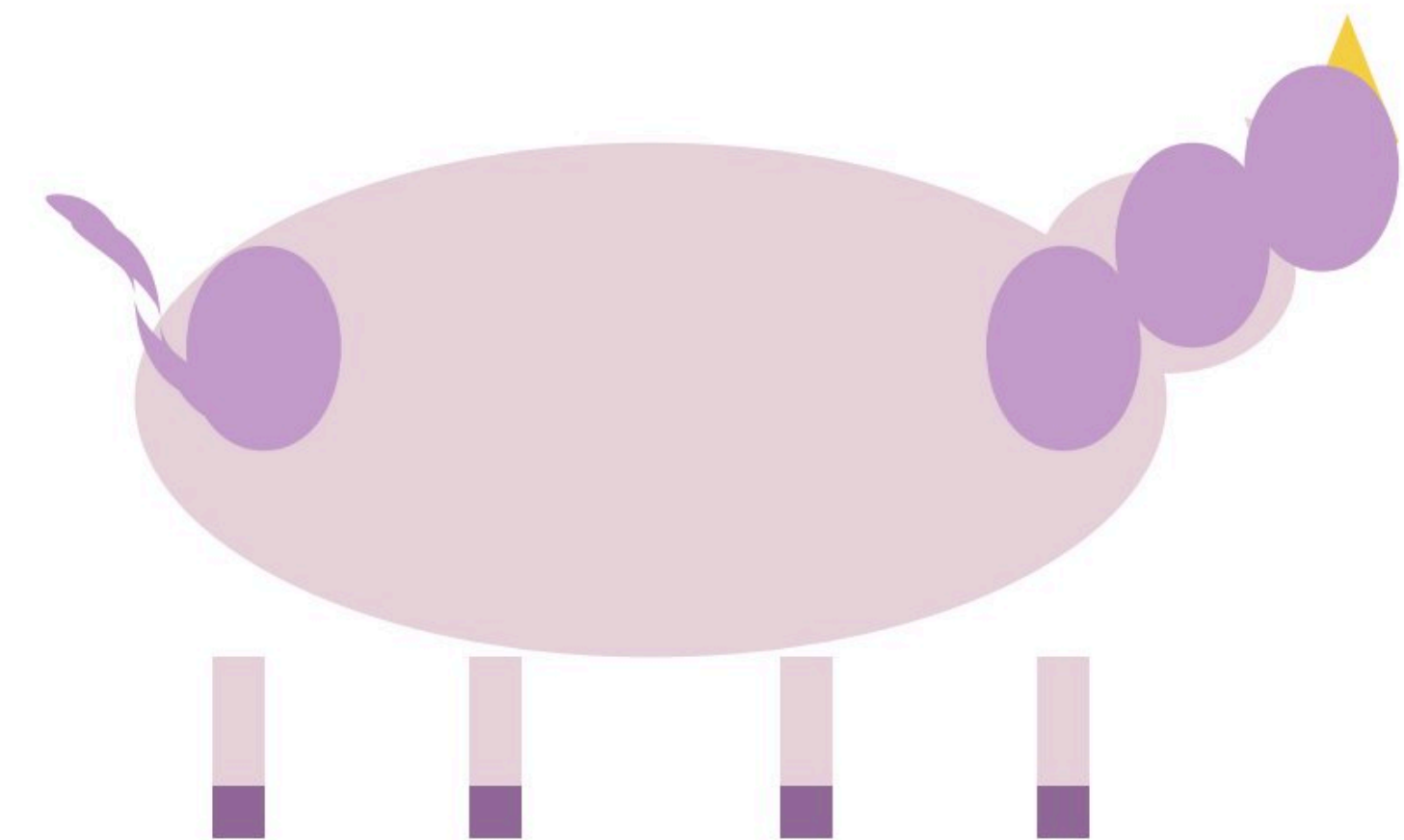
- **Smaller models** trained on **1.4T**, high-quality & publicly available data
- The models (LLaMA 2) are public
- “LLaMA-13B outperforms GPT-3 (175B) on most benchmarks, and LLaMA-65B is competitive with the best models, Chinchilla-70B and PaLM-540B”

GPT-4

- GPT-4: a **multi-modal** model capable of processing image and text inputs and producing text outputs.
- Model size and training details unknown
- Can process up to **32k** context size

Prompt: Draw a unicorn in TiKZ.

GPT-4: [Produces \LaTeX compiling to following picture.]



GPT-4

GPT-4 visual input example, Moar Layers:

User Can you explain why this is funny. Think about it step-by-step.



GPT-4 The comic is satirizing the difference in approaches to improving model performance between statistical learning and neural networks.

In statistical learning, the character is shown to be concerned with overfitting and suggests a series of complex and technical solutions, such as minimizing structural risk, reworking the loss function, and using a soft margin.

In contrast, the neural networks character simply suggests adding more layers to the model. This is often seen as a common solution to improving

GPT-4 visual input example, Extreme Ironing:

User What is unusual about this image?



Source: <https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg>

GPT-4 The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.