

# Natural Language Processing

AI51701/CSE71001

Lecture 17

11/16/2023

Instructor: Taehwan Kim

# Announcement

- ❑ Assignment 3 will be released early next week
  - You will be notified when it is out
  
- ❑ Final project
  - *Tentative Plan:* report due on 12/6,  
presentations on 12/7 and 12/12

# Prompting

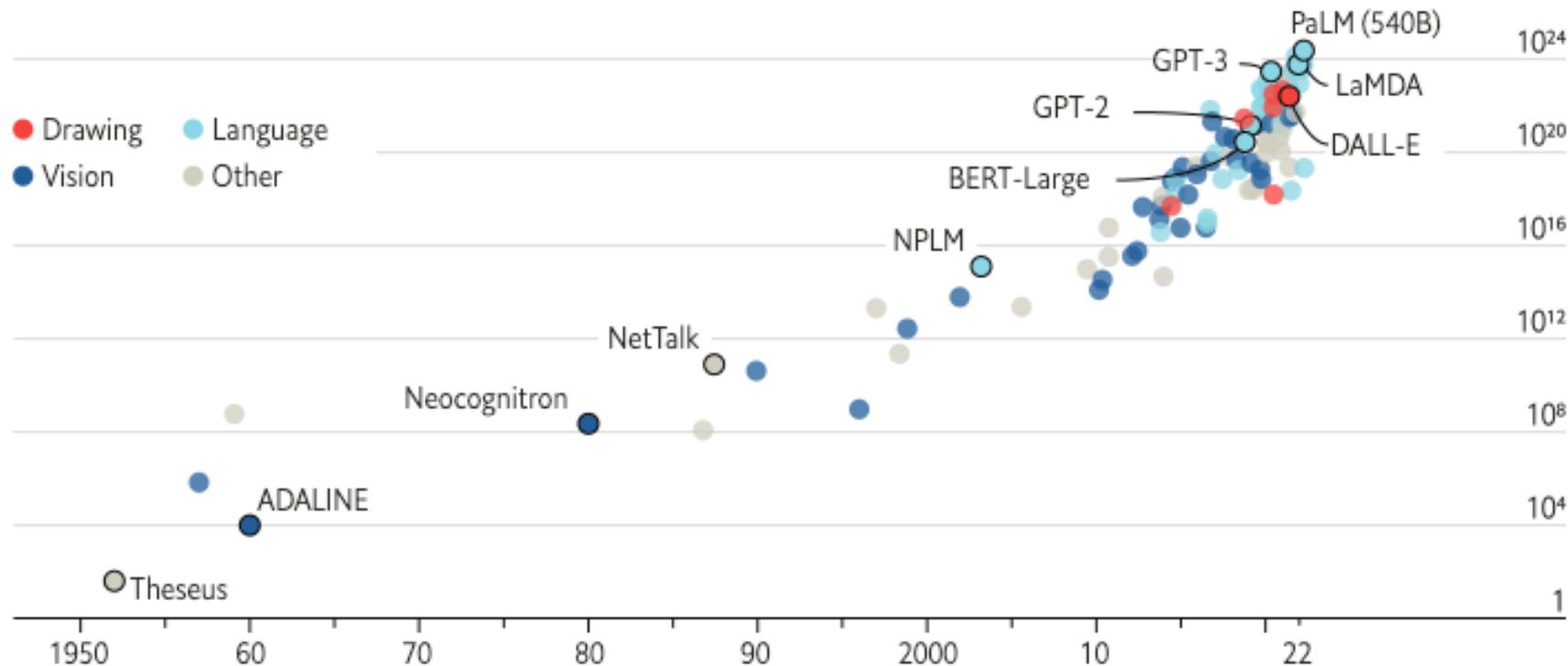
(Adapted from Jessy Mu's slides in Stanford 224n)

# Larger and larger models

## The blessings of scale

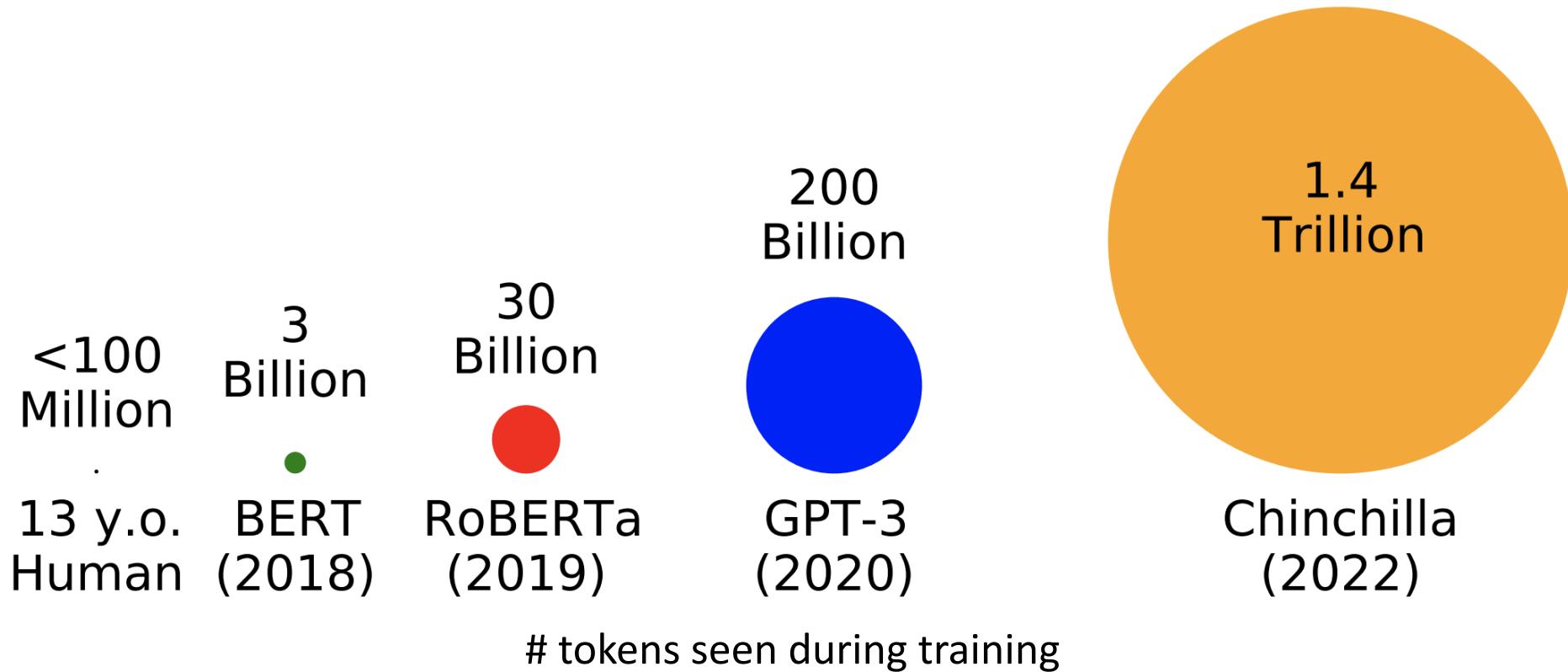
AI training runs, estimated computing resources used

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

# Trained on more and more data



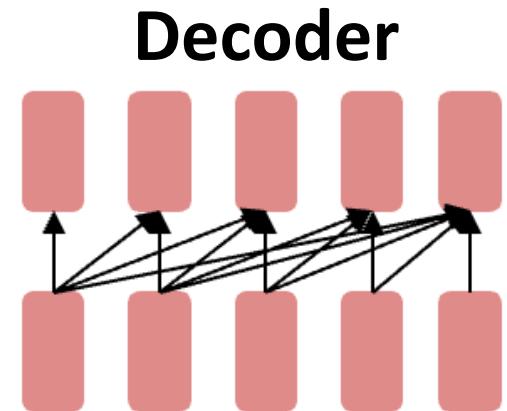
## What kinds of things does pretraining learn? (review)

- 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]
- I was thinking about the sequence that goes  
1, 1, 2, 3, 5, 8, 13, 21, \_\_\_\_\_ [some basic arithmetic]

# Zero-Shot (ZS) and Few-Shot (FS) In-Context Learning

# Emergent abilities of large language models: GPT (2018)

- Let's revisit the Generative Pretrained Transformer (GPT) models from OpenAI as an example:
- GPT (117M parameters; Radford et al., 2018)
  - Transformer decoder with 12 layers.
  - Trained on BooksCorpus: over 7000 unique books (4.6GB text).
- Showed that language modeling at scale can be an effective pretraining technique for downstream tasks like natural language inference.



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

# Emergent abilities of large language models: GPT-2 (2019)

- Let's revisit the Generative Pretrained Transformer (GPT) models from OpenAI as an example:
- GPT-2 (1.5B parameters; Radford et al., 2019)
  - Same architecture as GPT, just bigger (117M -> 1.5B)
  - But trained on **much more data**: 4GB -> 40GB of internet text data (WebText)
    - Scrape links posted on Reddit w/ at least 3 upvotes (rough proxy of human quality)

---

**Language Models are Unsupervised Multitask Learners**

---

# Emergent zero-shot learning

- One key emergent ability in GPT-2 is **zero-shot learning**: the ability to do many tasks with **no examples**, and **no gradient updates**, by simply:
  - Specifying the right sequence prediction problem (e.g. question answering):

Passage: Tom Brady... Q: Where was Tom Brady born? A: ...

- Comparing probabilities of sequences (e.g. Winograd Schema Challenge [Levesque, 2011]):

The cat couldn't fit into the hat because it was too big.  
Does it = the cat or the hat?

≡ Is  $P(\dots \text{because } \mathbf{\text{the cat}} \text{ was too big}) \geq P(\dots \text{because } \mathbf{\text{the hat}} \text{ was too big})$ ?

# Emergent zero-shot learning

- ❑ GPT-2 beats SoTA on language modeling benchmarks with no task-specific fine-tuning

*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 \_\_\_\_\_. **LAMBADA** (language modeling w/ long discourse dependencies)

*Target word:* Gabriel

[Paperno et al., 2016]

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14
117M	<b>35.13</b>	45.99	<b>87.65</b>	<b>83.4</b>	<b>29.41</b>
345M	<b>15.60</b>	55.48	<b>92.35</b>	<b>87.1</b>	<b>22.76</b>
762M	<b>10.87</b>	<b>60.12</b>	<b>93.45</b>	<b>88.0</b>	<b>19.93</b>
1542M	<b>8.63</b>	<b>63.24</b>	<b>93.30</b>	<b>89.05</b>	<b>18.34</b>

[Radford et al., 2019]

# Emergent abilities of large language models: GPT-3 (2020)

- GPT-3 (175B parameters; Brown et al., 2020)
  - Another increase in size (1.5B -> **175B**)
  - and data (40GB -> **over 600GB**)

---

## Language Models are Few-Shot Learners

---

Tom B. Brown\*

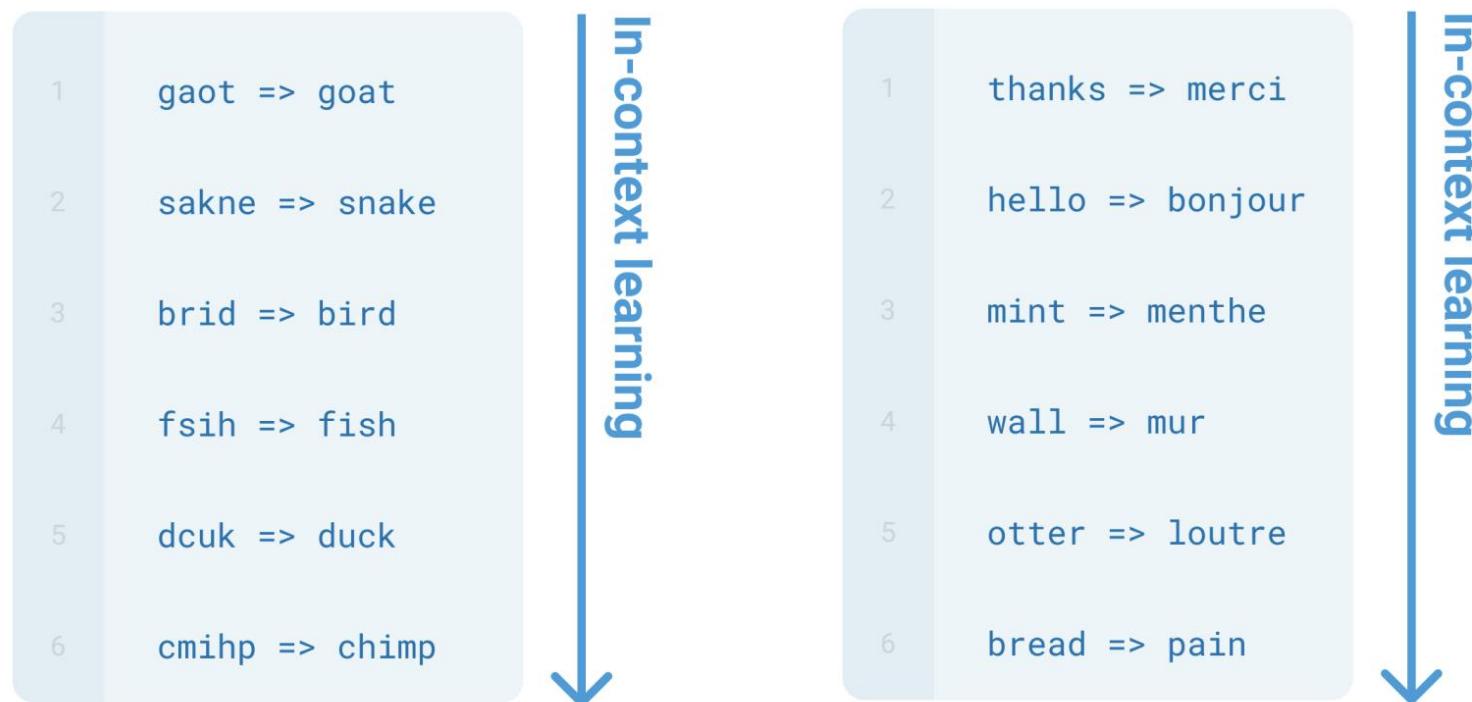
Benjamin Mann\*

Nick Ryder\*

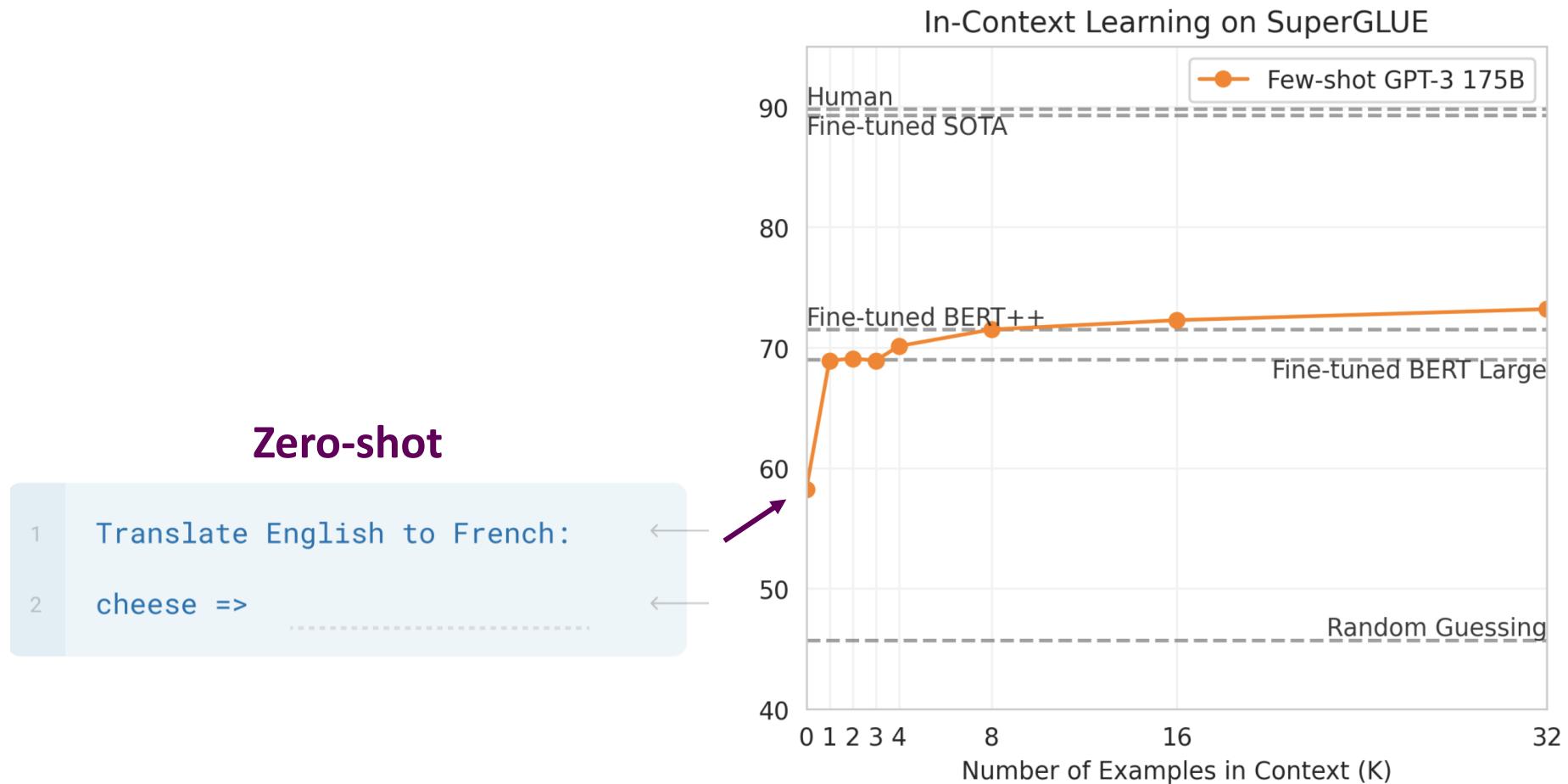
Melanie Subbiah\*

# Emergent few-shot learning

- Specify a task by simply **prepend**ing examples of the task before your example
- Also called **in-context learning**, to stress that *no gradient updates* are performed when learning a new task



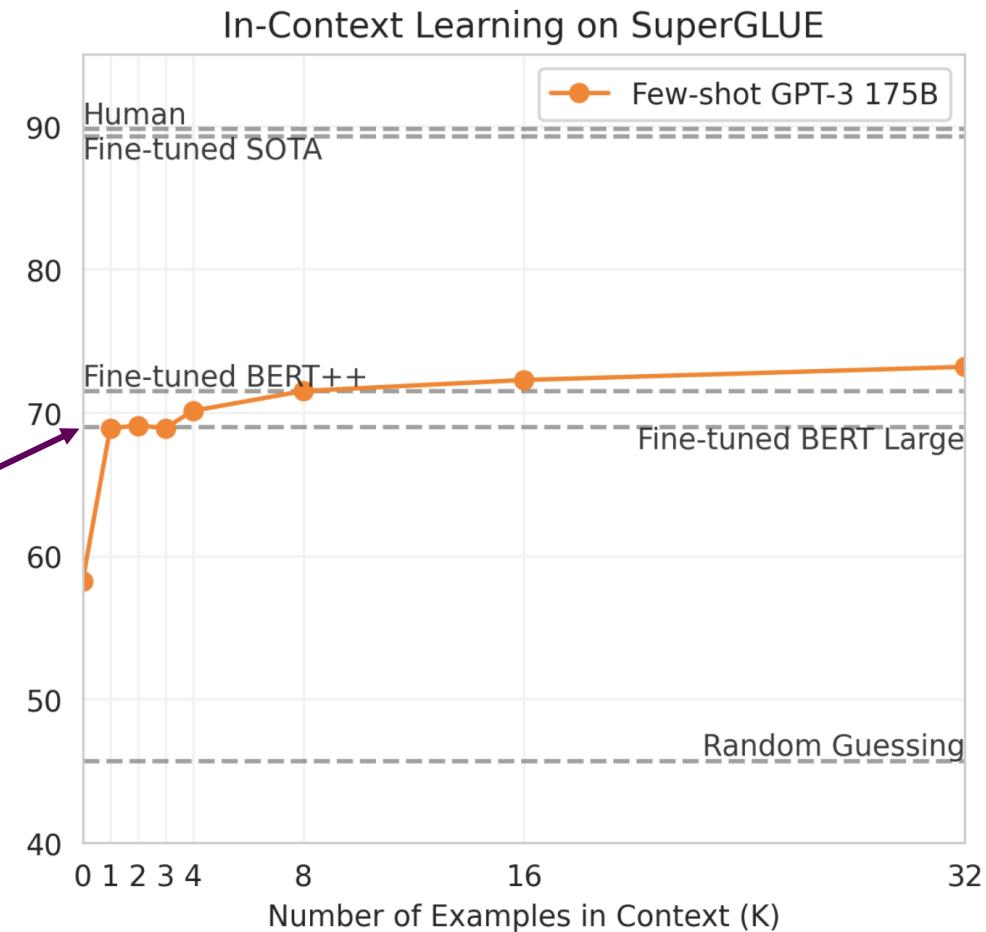
# Emergent few-shot learning



# Emergent few-shot learning

## One-shot

```
1 Translate English to French:  
2 sea otter => loutre de mer  
3 cheese =>
```



# Emergent few-shot learning

**Few-shot**

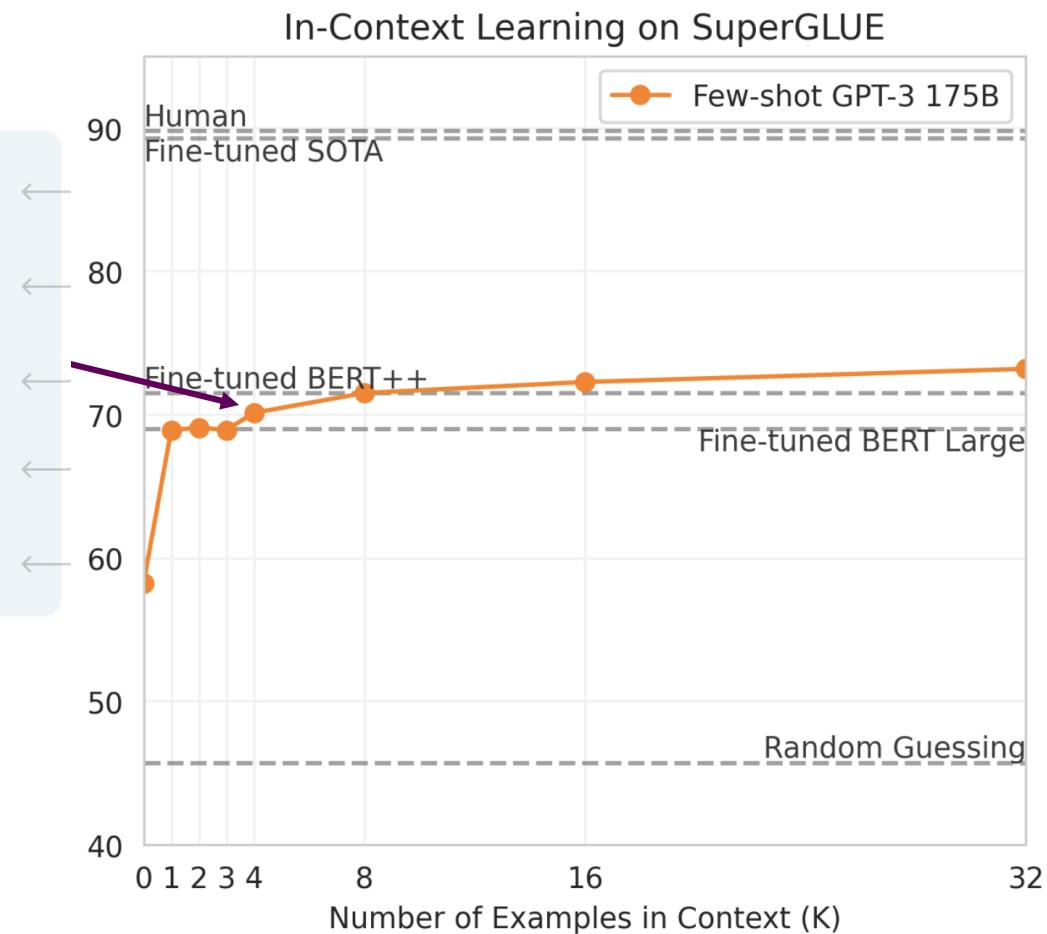
1 Translate English to French:

2 sea otter => loutre de mer

3 peppermint => menthe poivrée

4 plush girafe => girafe peluche

5 cheese => .....



# Few-shot learning is an emergent property of model scale

Cycle letters:

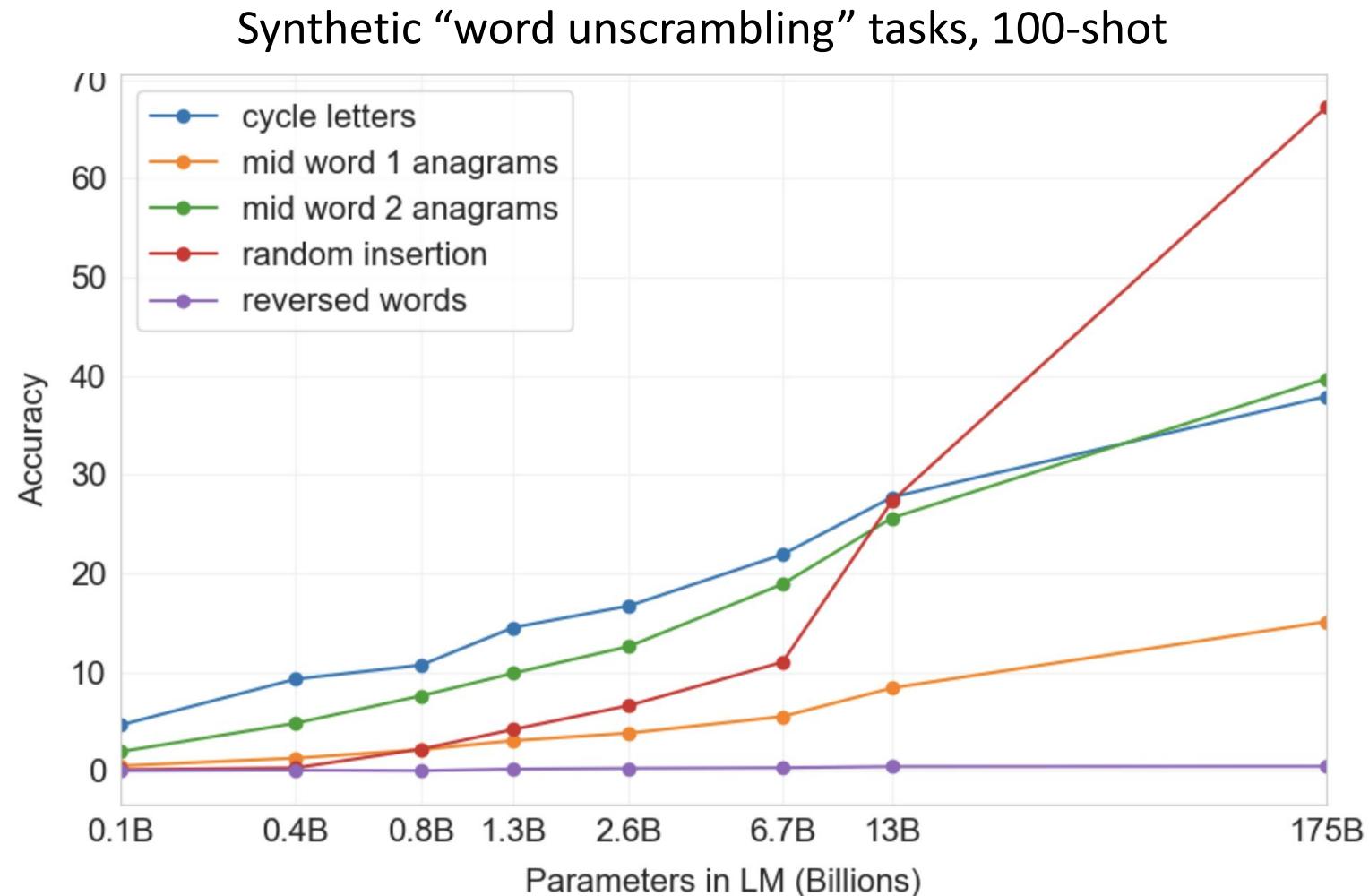
pleap ->  
apple

Random insertion:

a.p!p/l!e ->  
apple

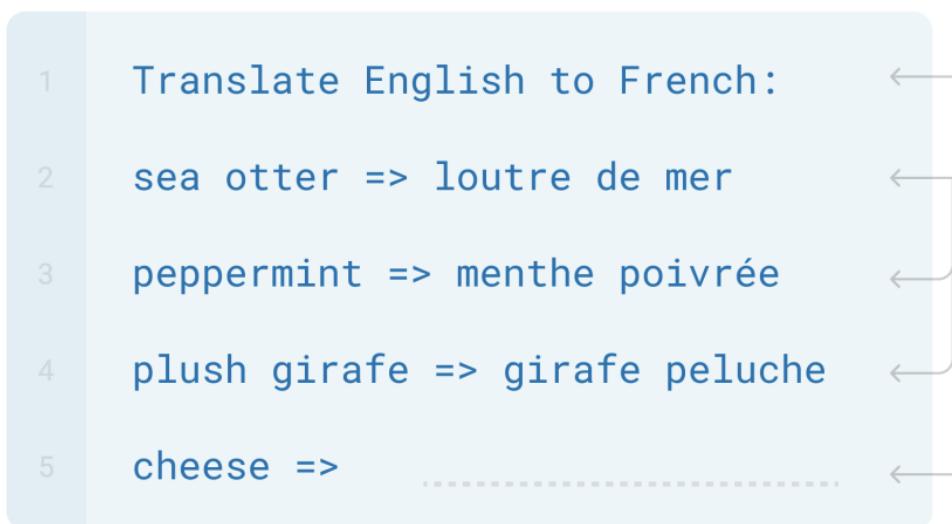
Reversed words:

elppa ->  
apple



# New methods of “prompting” LMs

## Zero/few-shot prompting



## Traditional fine-tuning



[Brown et al., 2020]

## Limits of prompting for harder tasks?

- Some tasks seem too hard for even large LMs to learn through prompting alone.  
Especially tasks involving richer, multi-step reasoning.  
(Humans struggle at these tasks too!)

$$19583 + 29534 = 49117$$

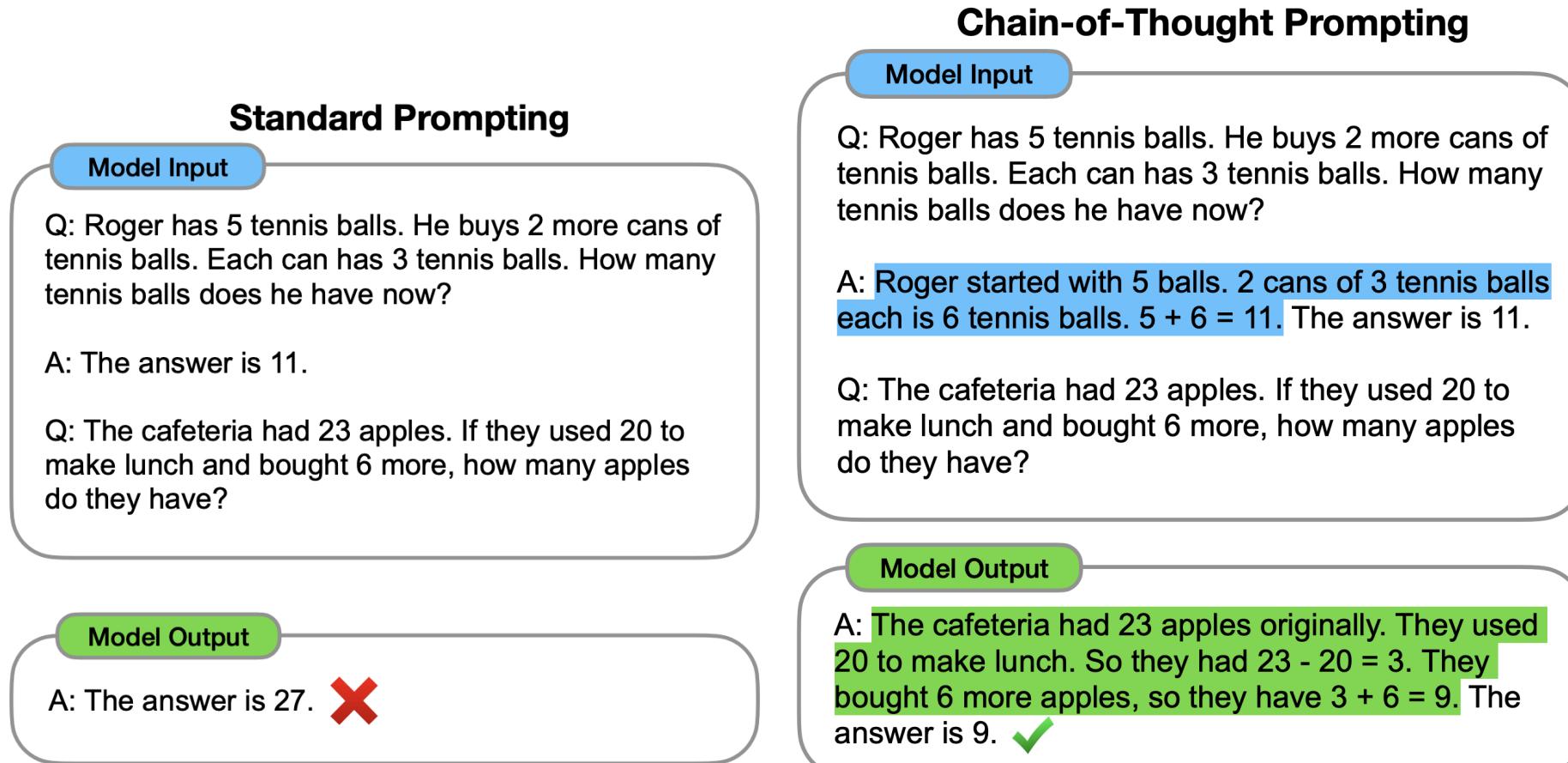
$$98394 + 49384 = 147778$$

$$29382 + 12347 = 41729$$

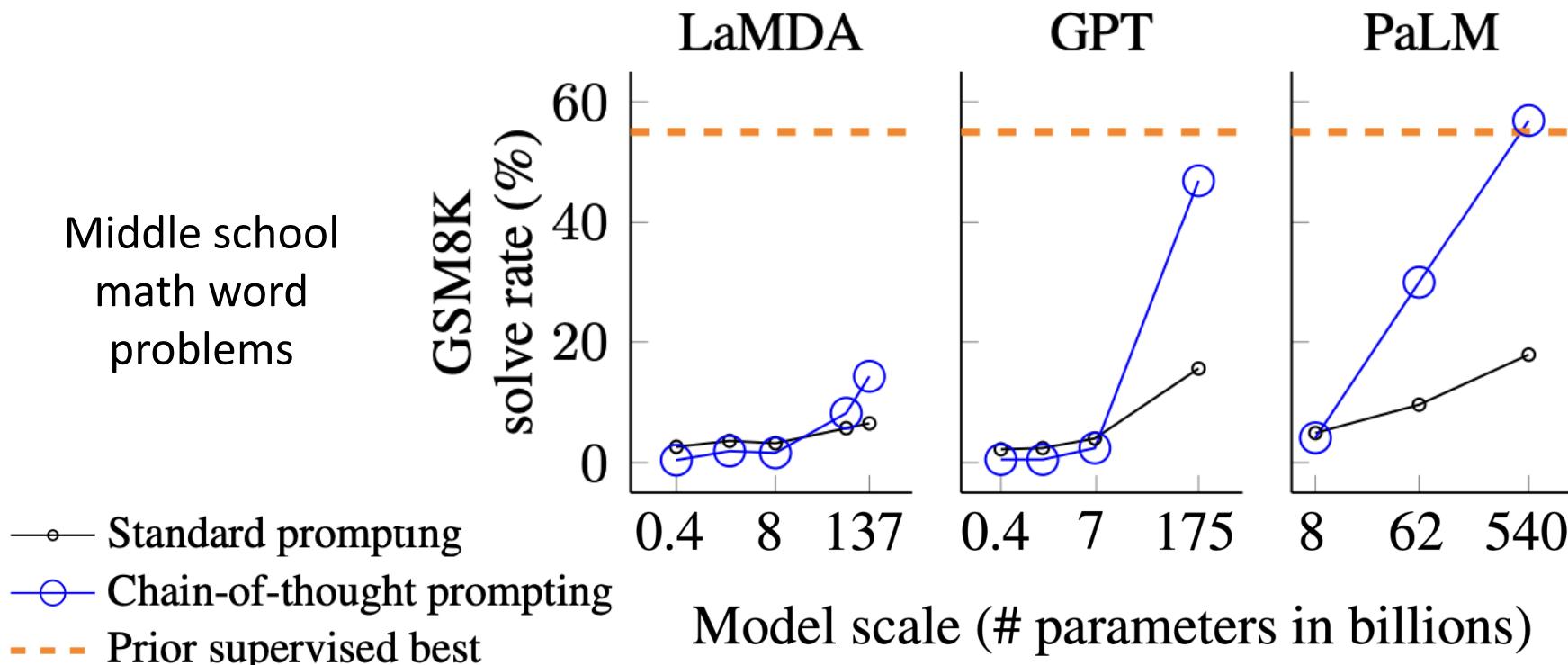
$$93847 + 39299 = ?$$

**Solution:** change the prompt!

# Chain-of-thought prompting

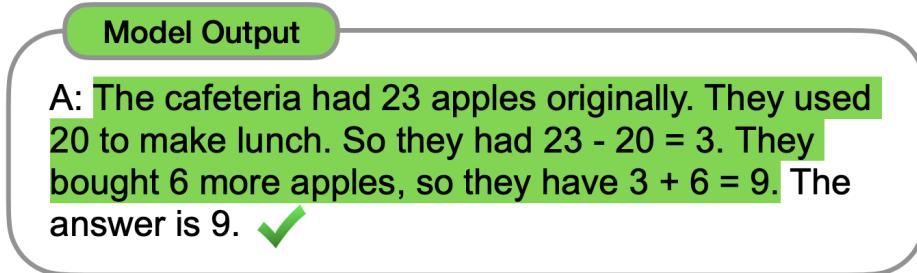
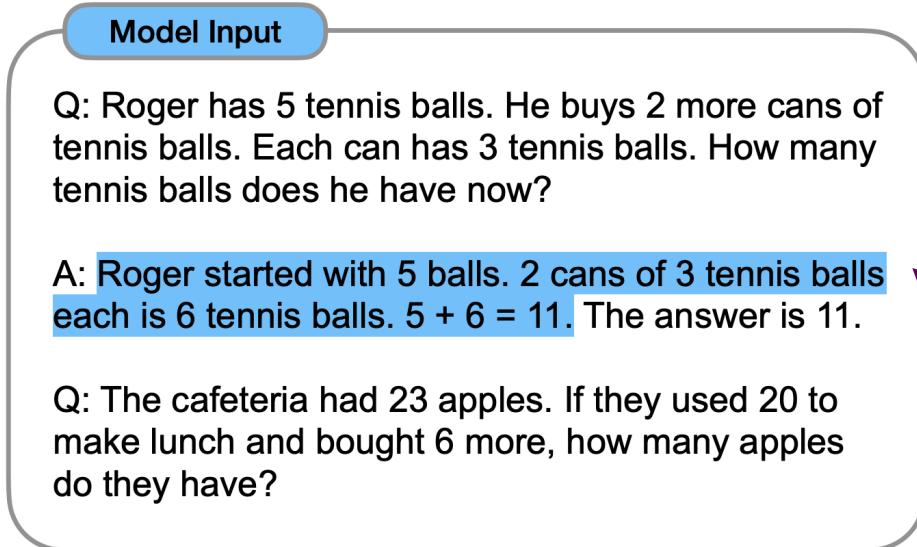


# Chain-of-thought prompting is an emergent property of model scale



(Wei et al., 2022; also see Nye et al., 2021)

# Chain-of-thought prompting



Do we even need examples of reasoning?  
Can we just ask the model to reason through things?

# Zero-shot chain-of-thought prompting

## Model 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. ✓

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.** There are 16 balls in total. Half of the balls are golf balls. That means there are 8 golf balls. Half of the golf balls are blue. That means there are 4 blue golf balls. ✓

# Zero-shot chain-of-thought prompting

	MultiArith	GSM8K
<b>Zero-Shot</b>	<b>17.7</b>	<b>10.4</b>
Few-Shot (2 samples)	33.7	15.6
Few-Shot (8 samples)	33.8	15.6
<b>Zero-Shot-CoT</b>	<b>Greatly outperforms → 78.7</b>	<b>40.7</b>
Few-Shot-CoT (2 samples)	zero-shot 84.8	41.3
Few-Shot-CoT (4 samples : First) (*1)	89.2	-
Few-Shot-CoT (4 samples : Second) (*1)	<b>Manual CoT → 90.5</b>	-
Few-Shot-CoT (8 samples)	still better 93.0	48.7

# Zero-shot chain-of-thought prompting

No.	Category	Zero-shot CoT Trigger Prompt	Accuracy
1	LM-Designed	Let's work this out in a step by step way to be sure we have the right answer.	<b>82.0</b>
2	Human-Designed	Let's think step by step. (*1)	78.7
3		First, (*2)	77.3
4		Let's think about this logically.	74.5
5		Let's solve this problem by splitting it into steps. (*3)	72.2
6		Let's be realistic and think step by step.	70.8
7		Let's think like a detective step by step.	70.3
8		Let's think	57.5
9		Before we dive into the answer,	55.7
10		The answer is after the proof.	45.7
-		(Zero-shot)	17.7

## Summary: Zero-Shot (ZS) and Few-Shot (FS) In-Context Learning

- ❑ + No finetuning needed, prompt engineering (e.g. CoT) can improve performance
- ❑ – Limits to what you can fit in context
- ❑ – Complex tasks will probably need gradient steps

# Instruction finetuning

# Language modeling ≠ assisting users

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.

- ❑ Language models are not aligned with user intent

# Language modeling ≠ assisting users

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.

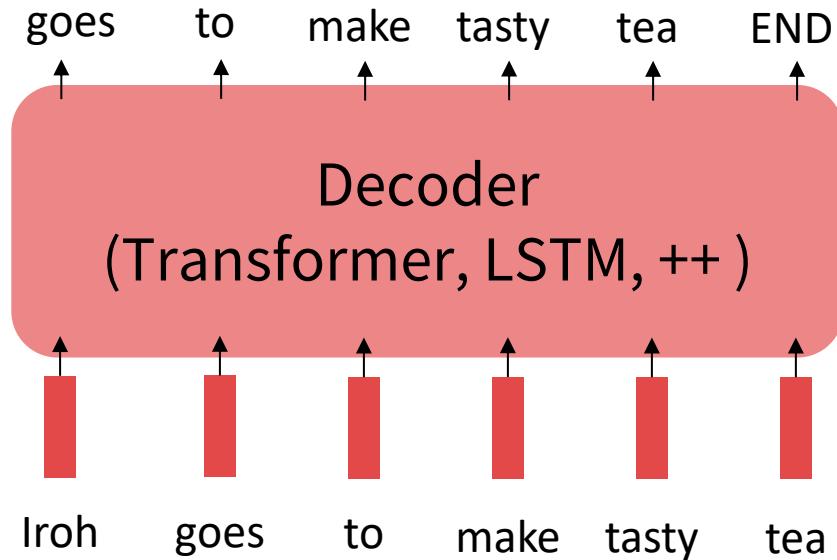
- Language models are not aligned with user intent
- Finetuning to the rescue!

# The Pretraining / Finetuning Paradigm (review)

- ❑ Pretraining can improve NLP applications by serving as parameter initialization.

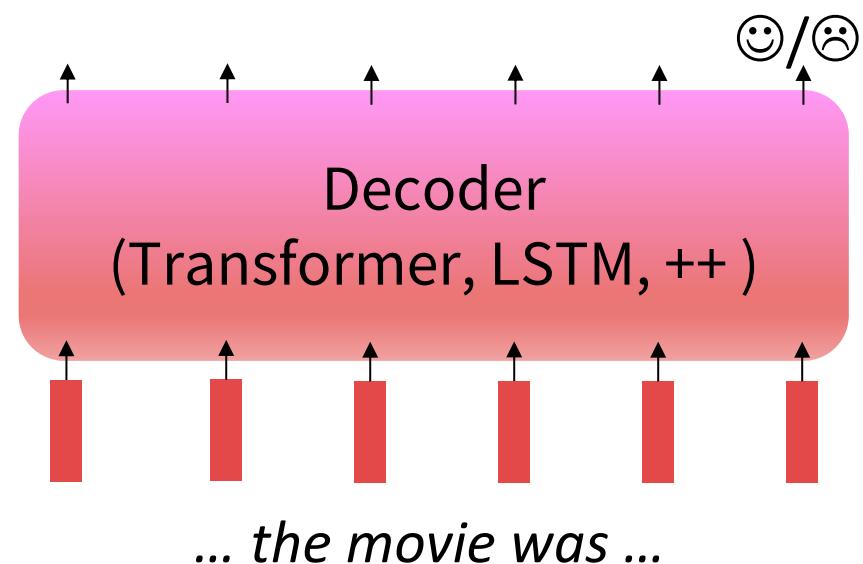
## Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



## Step 2: Finetune (on your task)

Not many labels; adapt to the task!

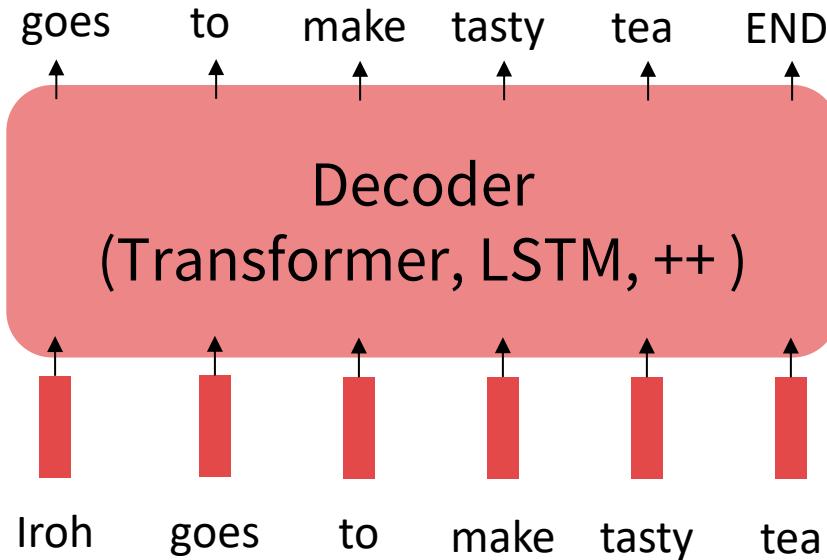


# Scaling up finetuning

- ❑ Pretraining can improve NLP applications by serving as parameter initialization.

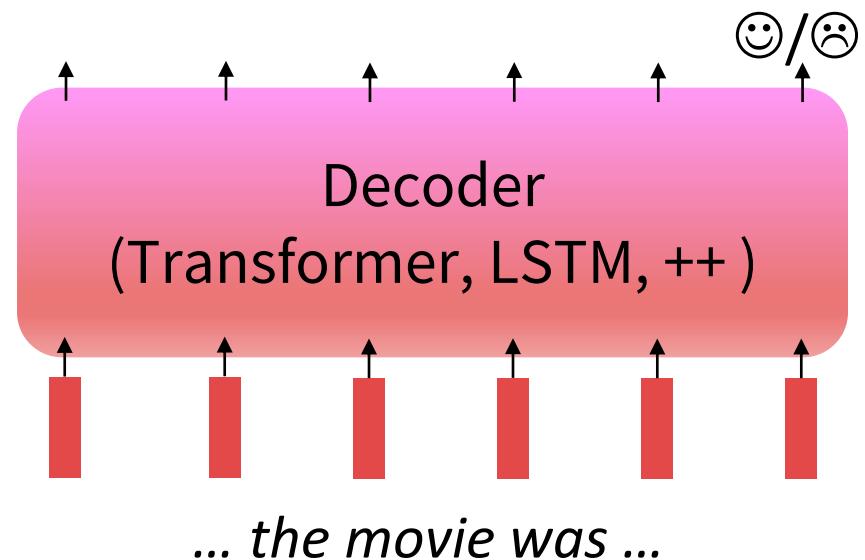
## Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



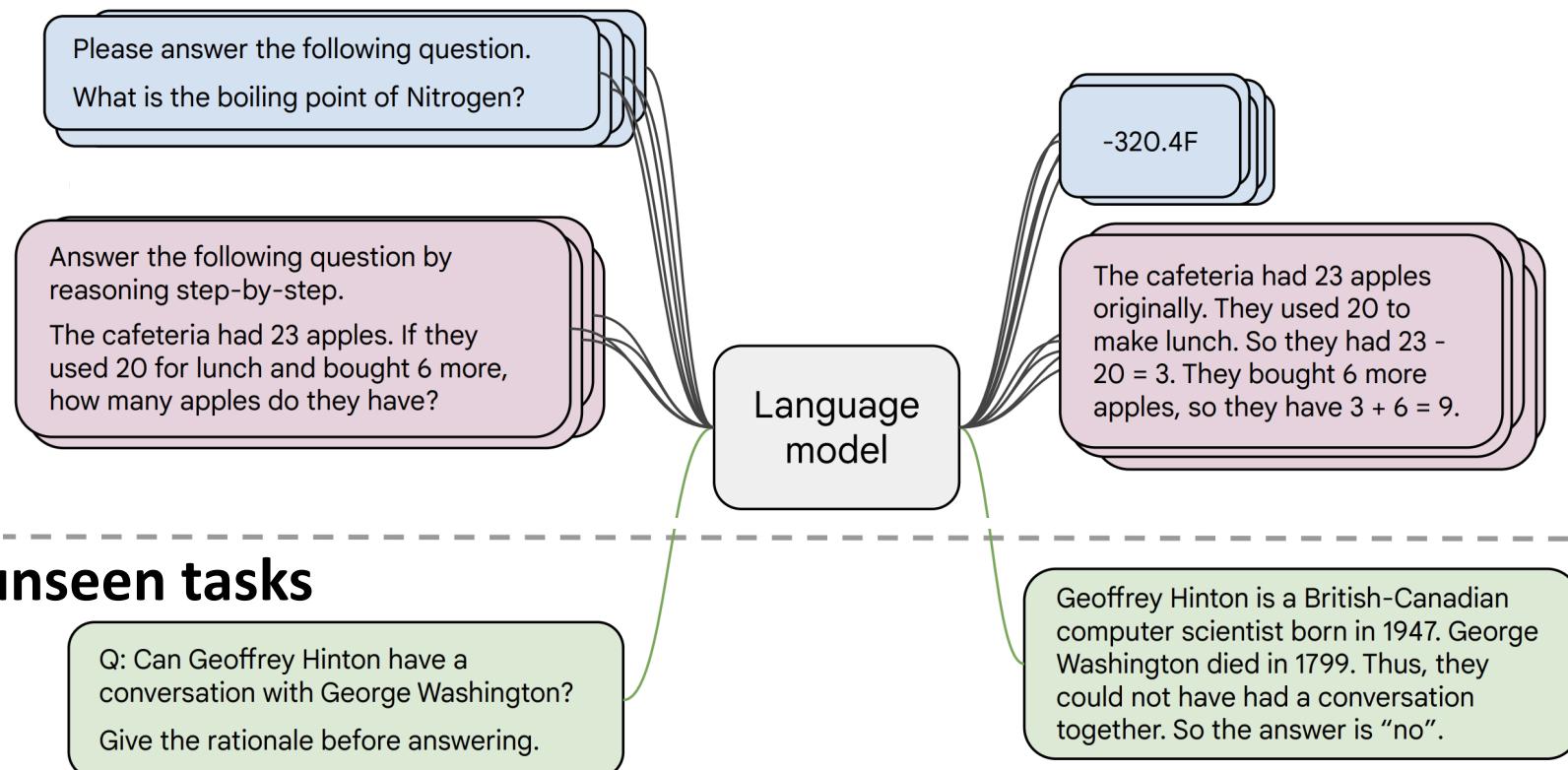
## Step 2: Finetune (on many tasks)

Not many labels; adapt to the tasks!



# Instruction finetuning

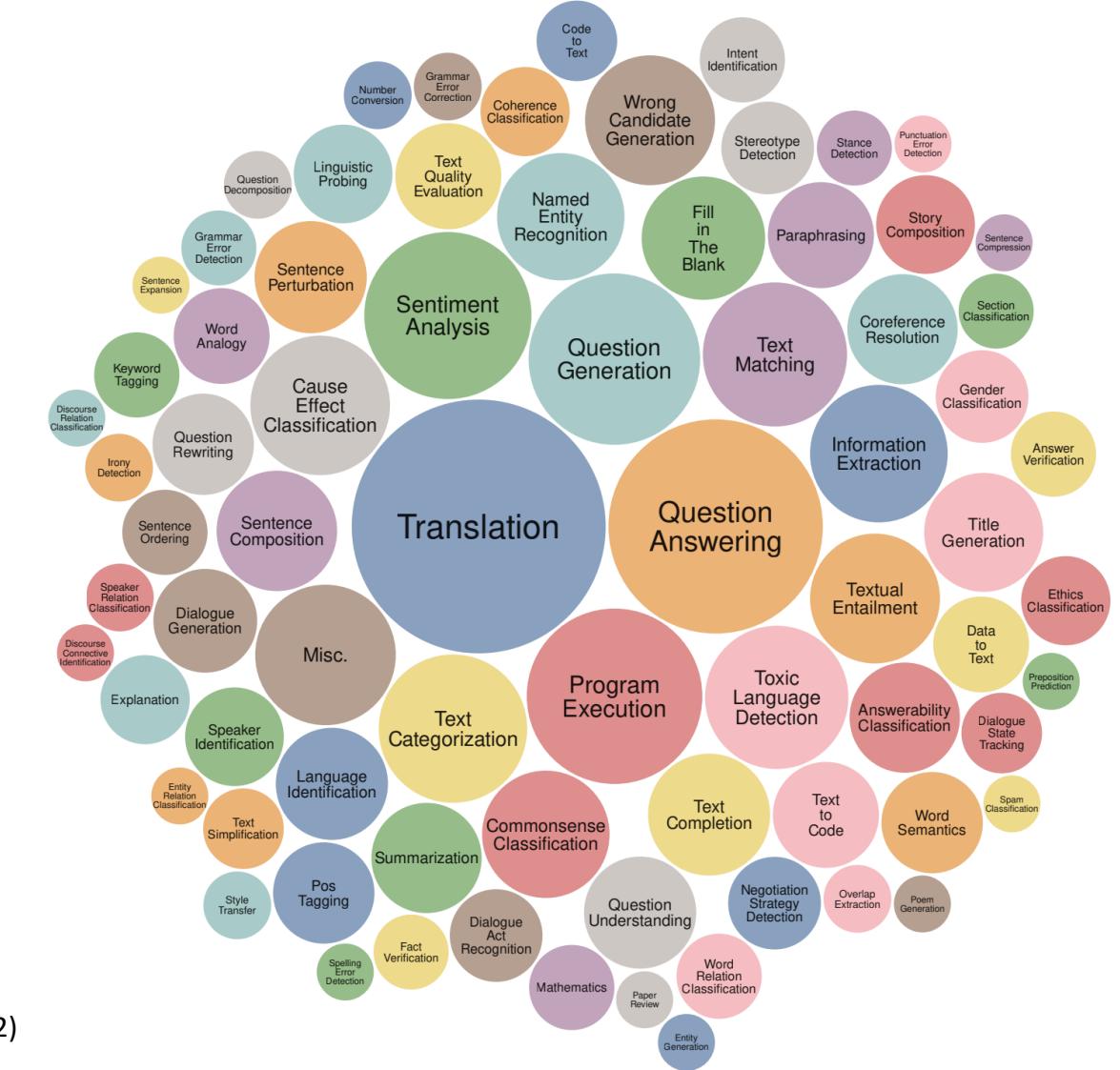
- ❑ Collect examples of (instruction, output) pairs across many tasks and finetune an LM



- ❑ Evaluate on **unseen** tasks

# Instruction finetuning (pretraining?)

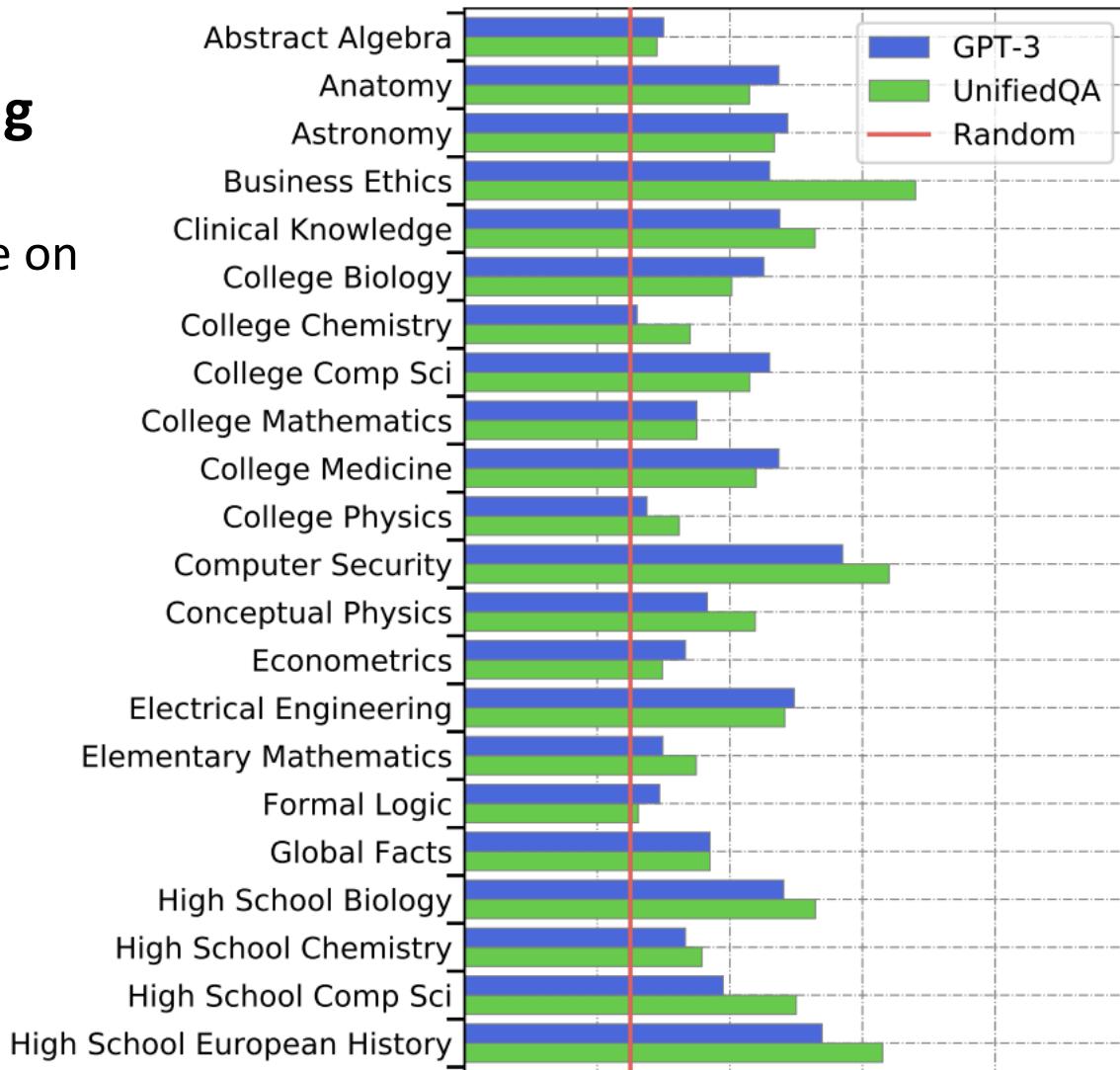
- ❑ As is usually the case, **data + model scale** is key for this to work!
- ❑ For example, the **Super-NaturalInstructions** dataset contains over 1.6K tasks, 3M+ examples
  - Classification, sequence tagging, rewriting, translation, QA...
- ❑ Q: how do we evaluate such a model?



# Aside: new benchmarks for multitask LMs

## □ Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021)

- New benchmarks for measuring LM performance on 57 diverse *knowledge intensive* tasks



# Aside: new benchmarks for multitask LMs

## □ BIG-Bench (Srivastava et al., 2022)

- 200+ tasks, spanning:



[https://github.com/google/BIG-bench/blob/main/bigbench/benchmark\\_tasks/README.md](https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/README.md)

## BEYOND THE IMITATION GAME: QUANTIFYING AND EXTRAPOLATING THE CAPABILITIES OF LANGUAGE MODELS

### Alphabetic author list:<sup>\*</sup>

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Asklund, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullikandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özütür, Behnam Heydayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Cameron Dia, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chittar Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Gabaceas, Damien Silco, Dan Kilday, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moscifú González, Daniela Perszyk, Danny Hernandez, Danji Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Klejny, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Dileyan Misra, Dilyan Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellis Pavlich, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Ertuğrul Erdem, Erie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Collet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovich-López, Gregor Betz, Guy Gur-Ari, Hana Galijasević, Hannah Kim, Hannnah Rashkin, Hannnah Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaan Jumelet, Jack Geissinger, Jackson Kermion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Tang, Jillian Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Berant, Jörg Frohberg, Joe Rozen, Jose Hernandez-Orallo, Joseph Bouledjoud, Joseph Jones, Joshua B. Lam, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Kartik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiaffullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Larisa Reynolds, Leo Gao, Li Zhang, Linda Dugan, Lianhua Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfü Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez Quintana, Marie Tolkiene, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Matýás Schubert, Medina Orduna-Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starratt, Michael Struble, Michał Swiderski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimes Xu, Miras Suzgun, Mo Tiwari, Mohit Bansal, Moin Aminasari, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nikita Nangia, Niklas Deckers, Niklas Muenmighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pi Liang, Paul Viol, Pegah Alipoormolabahshi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phi Mon Huit, Pinyu Hwang, Piotr Milkowski, Piyush Patil, Pouya Pezeshpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Robin Banjade, Rachel Etta Rudolph, Rafee Gabriel, Rahel Habacker, Ramón Risco Delgado, Raphaël Millière, Rhythym Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbie Raymakers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sittleve, Ronan LeBras, Rosanna Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Ryhan Yang, Sabih Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Guetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shan, Gu, Shubh Pachigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima (Shammie) Debnath, Siamak Shakeri, Simon Thormeyer, Simeone Melizi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Bidenman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Mishhergi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tat Schuster, Ta Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschid, Thomas Phan, Tianle Wang, Tiberius Nkonyili, Timo Schick, Timofei Kornev, Timothy Tellegen-Lawton, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikanth, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, Ziyi Wu

## Aside: new benchmarks for multitask LMs

□ BIG-Bench (Srivastava et al., 2022)

- 200+ tasks, spanning:

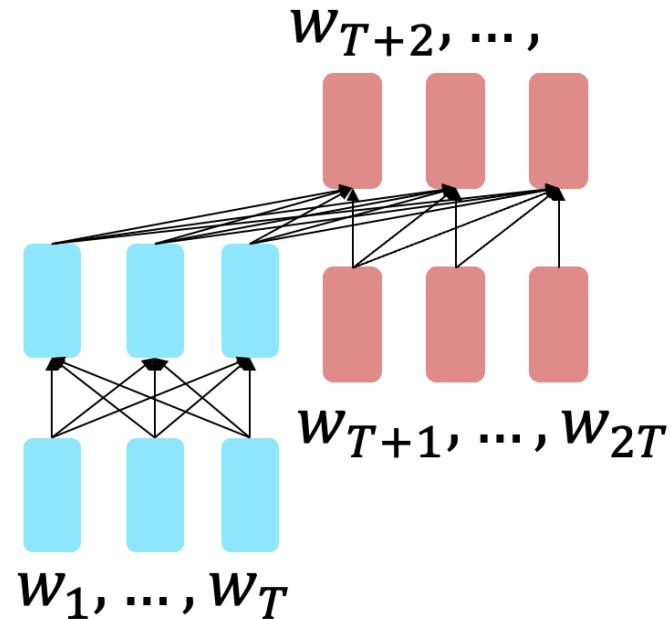


# Kanji ASCII Art to Meaning

This subtask converts various kanji into ASCII art and has the language model guess their meaning from the ASCII art.

# Summary: Instruction finetuning

- Recall the T5 encoder-decoder model pretrained on the span corruption task
- **Flan-T5** (Chung et al., 2020): T5 models finetuned on 1.8K additional tasks



(Chung et al., 2022)

Params	Model	BIG-bench + MMLU avg (normalized)
80M	T5-Small	-9.2
	Flan-T5-Small	-3.1 <b>(+6.1)</b>
250M	T5-Base	-5.1
	Flan-T5-Base	6.5 <b>(+11.6)</b>
780M	T5-Large	-5.0
	Flan-T5-Large	13.8 <b>(+18.8)</b>
3B	T5-XL	-4.1
	Flan-T5-XL	19.1 <b>(+23.2)</b>
11B	T5-XXL	-2.9
	Flan-T5-XXL	23.7 <b>(+26.6)</b>

**Bigger model = bigger  $\Delta$**

# Instruction finetuning

## 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.

## Before instruction finetuning

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)

Highly recommend trying FLAN-T5 out to get a sense of its capabilities:  
<https://huggingface.co/google/flan-t5-xxl>

(Chung et al., 2022)

# Instruction finetuning

## 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.

## After instruction finetuning

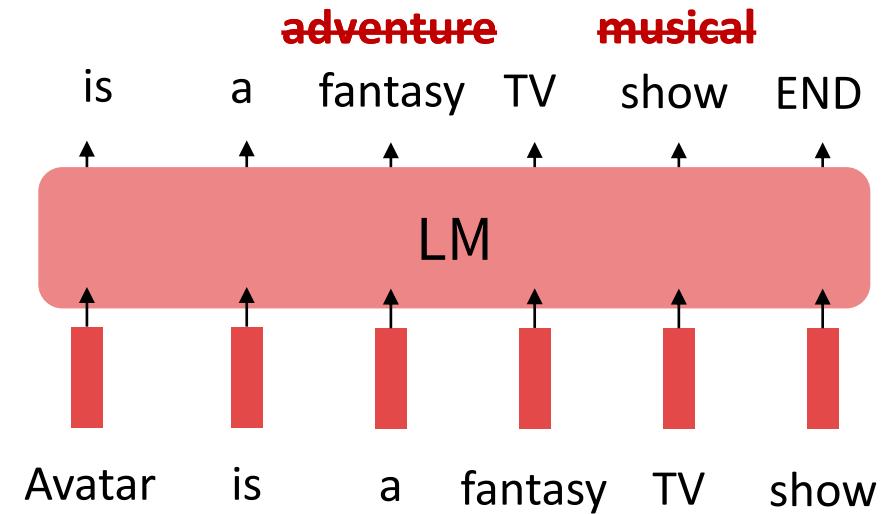
The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). 

Highly recommend trying FLAN-T5 out to get a sense of its capabilities:  
<https://huggingface.co/google/flan-t5-xxl>

(Chung et al., 2022)

# Limitations of instruction finetuning?

- ❑ One limitation of instruction finetuning is obvious: it's **expensive** to collect ground-truth data for tasks.
- ❑ But there are other, subtler limitations too. Can you think of any?
- ❑ **Problem 1:** tasks like open-ended creative generation have no right answer.
  - *Write me a story about a dog and her pet grasshopper.*
- ❑ **Problem 2:** language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
- ❑ Even with instruction finetuning, there is a mismatch between the LM objective and the objective of “satisfy human preferences”!
- ❑ Can we **explicitly attempt to satisfy human preferences?**



## Summary: Instruction finetuning

- + Simple and straightforward, generalize to unseen tasks
- Collecting demonstrations for so many tasks is expensive
- Mismatch between LM objective and human preferences

# Reinforcement Learning from Human Feedback (RLHF)

# Optimizing for human preferences

- ❑ Let's say we were training a language model on some task (e.g. summarization).
- ❑ For each LM sample  $s$ , imagine we had a way to obtain a *human reward* of that summary:  $R(s) \in \mathbb{R}$ , higher is better.

SAN FRANCISCO,  
California (CNN) --  
A magnitude 4.2  
earthquake shook the  
San Francisco  
...  
overturn unstable  
objects.

An earthquake hit  
San Francisco.  
There was minor  
property damage,  
but no injuries.

$$s_1 \\ R(s_1) = 8.0$$

The Bay Area has  
good weather but is  
prone to  
earthquakes and  
wildfires.

$$s_2 \\ R(s_2) = 1.2$$

- ❑ Now we want to maximize the expected reward of samples from our LM:

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

# Reinforcement learning to the rescue

- The field of **reinforcement learning (RL)** has studied these (and related) problems for many years now (Williams, 1992; Sutton and Barto, 1998)
- Circa 2013: resurgence of interest in RL applied to deep learning, game-playing (Mnih et al., 2013)
- But the interest in applying RL to modern LMs is an even newer phenomenon (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022). Why?
  - RL w/ LMs has commonly been viewed as very hard to get right (still is!)
  - Newer advances in RL algorithms that work for large neural models, including language models (e.g. PPO; (Schulman et al., 2017))

# Optimizing for human preferences

- How do we actually change our LM parameters  $\theta$  to maximize this?

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

- Let's try doing gradient ascent!

$$\theta_{t+1} := \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{\hat{s} \sim p_{\theta_t}(s)}[R(\hat{s})]$$

How do we estimate  
this expectation??

What if our reward  
function is non-  
differentiable??

- **Policy gradient** methods in RL (e.g., REINFORCE; (Williams, 1992)) give us tools for estimating and optimizing this objective.
- We'll describe a *very high-level* mathematical overview of the simplest policy gradient estimator, but a full treatment of RL is outside the scope of this course.

# A (very!) brief introduction to policy gradient/REINFORCE (Williams, 1992)

- We want to obtain

$$\nabla_{\theta} \mathbb{E}_{\hat{s} \sim p_{\theta}(s)} [R(\hat{s})] = \nabla_{\theta} \sum_s R(s) p_{\theta}(s) \stackrel{\text{(defn. of expectation)}}{=} \sum_s R(s) \nabla_{\theta} p_{\theta}(s) \stackrel{\text{(linearity of gradient)}}{=}$$

- Here we'll use a very handy trick known as the log-derivative trick. Let's try taking the gradient of  $\log p_{\theta}(s)$

$$\nabla_{\theta} \log p_{\theta}(s) = \frac{1}{p_{\theta}(s)} \nabla_{\theta} p_{\theta}(s) \quad \Rightarrow \quad \nabla_{\theta} p_{\theta}(s) = \nabla_{\theta} \log p_{\theta}(s) p_{\theta}(s)$$

(chain rule)

- Plug back in:

This is an expectation of this

$$\begin{aligned} \sum_s R(s) \nabla_{\theta} p_{\theta}(s) &= \sum_s p_{\theta}(s) R(s) \nabla_{\theta} \log p_{\theta}(s) \\ &= \mathbb{E}_{\hat{s} \sim p_{\theta}(s)} [R(\hat{s}) \nabla_{\theta} \log p_{\theta}(\hat{s})] \end{aligned}$$

# A (very!) brief introduction to policy gradient/REINFORCE (Williams, 1992)

- Now we have put the gradient “inside” the expectation, we can approximate this objective with Monte Carlo samples:

$$\nabla_{\theta} \mathbb{E}_{\hat{s} \sim p_{\theta}(s)} [R(\hat{s})] = \mathbb{E}_{\hat{s} \sim p_{\theta}(s)} [R(\hat{s}) \nabla_{\theta} \log p_{\theta}(\hat{s})] \approx \frac{1}{m} \sum_{i=1}^m R(s_i) \nabla_{\theta} \log p_{\theta}(s_i)$$

- This is why it’s called “reinforcement learning”: we reinforce good actions, increasing the chance they happen again.

- Giving us the update rule:

$$\theta_{t+1} := \theta_t + \alpha \frac{1}{m} \sum_{i=1}^m R(s_i) \nabla_{\theta_t} \log p_{\theta_t}(s_i)$$

If  $R$  is +++ Take gradient steps to maximize  $p_{\theta}(s_i)$

If  $R$  is --- Take steps to minimize  $p_{\theta}(s_i)$

- This is heavily simplified! There is a lot more needed to do RL w/ LMs. Can you see any problems with this objective?

# How do we model human preferences?

- ❑ Awesome: now for any **arbitrary, non-differentiable reward function**  $R(s)$ , we can train our language model to maximize expected reward.
- ❑ Not so fast! (Why not?)
- ❑ **Problem 1:** human-in-the-loop is expensive!
  - **Solution:** instead of directly asking humans for preferences, **model their preferences** as a separate (NLP) problem! (Knox and Stone, 2009)

An earthquake hit  
San Francisco.  
There was minor  
property damage,  
but no injuries.

$$s_1 \\ R(s_1) = 8.0 \quad \text{👤} \quad \text{💵💵}$$

The Bay Area has  
good weather but is  
prone to  
earthquakes and  
wildfires.

$$s_2 \\ R(s_2) = 1.2 \quad \text{👤} \quad \text{💵💵}$$

Train an LM  $RM_\phi(s)$  to  
predict human  
preferences from an  
annotated dataset, then  
optimize for  $RM_\phi$  instead.

# How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable (Phelps et al., 2015; Clark et al., 2018)

A 4.2 magnitude  
earthquake hit  
San Francisco,  
resulting in  
massive damage.

$s_3$

$$R(s_3) = \text{ } 4.1? \text{ } 6.6? \text{ } 3.2?$$

# How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
  - **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable (Phelps et al., 2015; Clark et al., 2018)

An earthquake hit  
San Francisco.  
There was minor  
property damage,  
but no injuries.

3

1.2

The Bay Area ... wildfires

$S_1$

Reward Model ( $RM_\phi$ )

1.2

A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.

1

S<sub>3</sub>

## Bradley-Terry [1952] paired comparison model

$$J_{RM}(\phi) = -\mathbb{E}_{(\textcolor{blue}{S^w}, \textcolor{red}{S^l}) \sim D} [\log \sigma(RM_\phi(\textcolor{blue}{S^w}) - RM_\phi(\textcolor{red}{S^l}))]$$

“winning” sample      “losing” sample

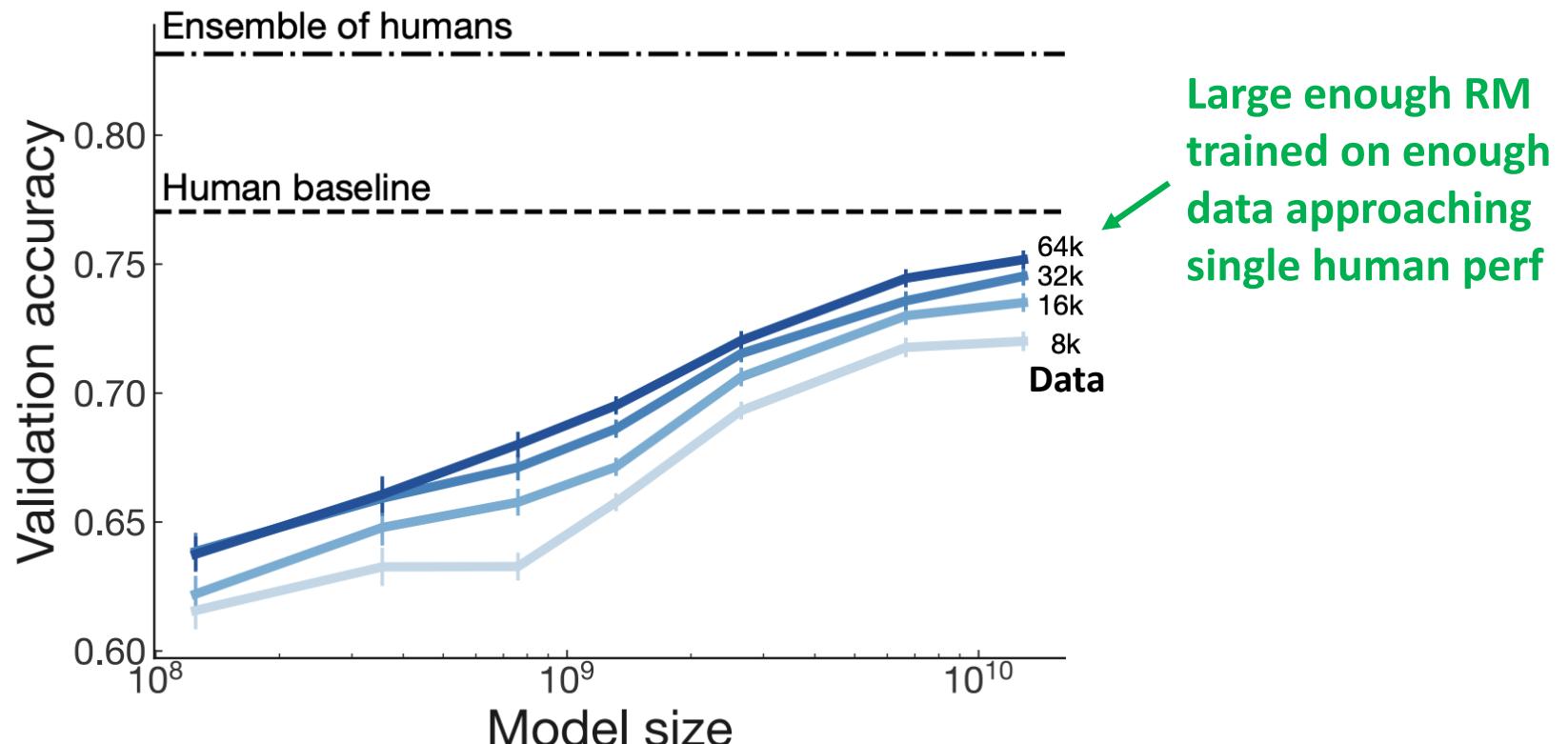
Bay Area has  
weather but is  
e to  
nquakes and  
fires.

S<sub>2</sub>

$s^w$  should score  
higher than  $s^l$

# Make sure your reward model works first!

- ❑ Evaluate RM on predicting outcome of held-out human judgments



(Stiennon et al., 2020)

# RLHF: Putting it all together (Christiano et al., 2017; Stiennon et al., 2020)

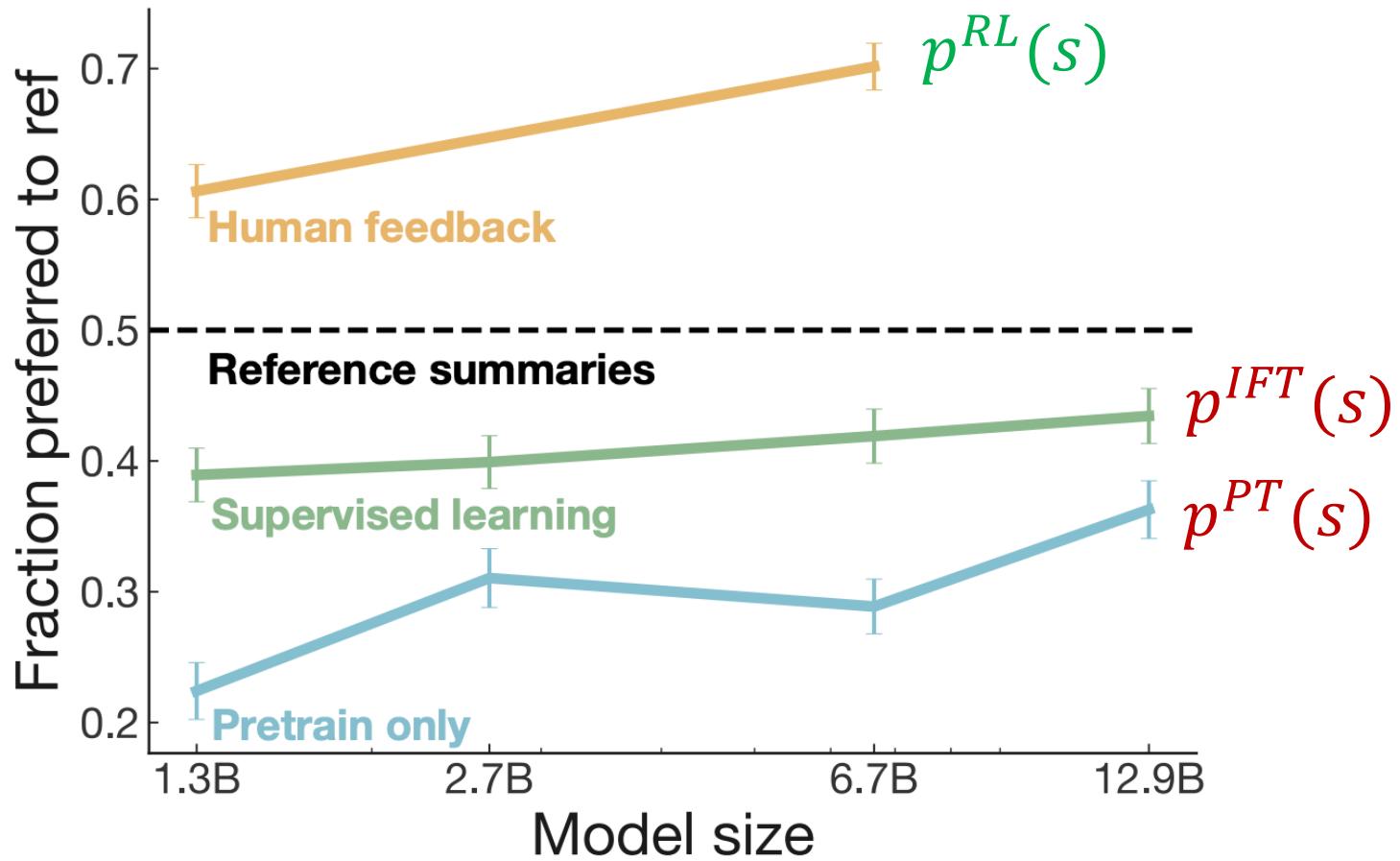
- Finally, we have everything we need:
  - A pretrained (possibly instruction-finetuned) LM  $p^{PT}(s)$
  - A reward model  $RM_{\phi}(s)$  that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function.
- Now to do RLHF:
  - Initialize a copy of the model  $p_{\theta}^{RL}(s)$ , with parameters  $\theta$  we would like to optimize
  - Optimize the following reward with RL:

$$R(s) = RM_{\phi}(s) - \beta \log \left( \frac{p_{\theta}^{RL}(s)}{p^{PT}(s)} \right)$$

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

This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler (KL)** divergence between  $p_{\theta}^{RL}(s)$  and  $p^{PT}(s)$ .

# RLHF provides gains over pretraining + finetuning



(Stiennon et al., 2020)

# InstructGPT: scaling up RLHF to tens of thousands of tasks

**30k  
tasks!**

Step 1

**Collect demonstration data,  
and train a supervised policy.**

A prompt is sampled from our prompt dataset.

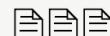


A labeler demonstrates the desired output behavior.



Some people went to the moon...

This data is used to fine-tune GPT-3 with supervised learning.



Step 2

**Collect comparison data,  
and train a reward model.**

A prompt and several model outputs are sampled.



A, B, C, D

Explain gravity... Explain war...



C Moon is natural satellite of...

D People went to the moon...

A labeler ranks the outputs from best to worst.



D > C > A = B

This data is used to train our reward model.



D > C > A = B

Step 3

**Optimize a policy against the reward model using reinforcement learning.**

A new prompt is sampled from the dataset.



The policy generates an output.



Once upon a time...

The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

r<sub>k</sub>

(Ouyang et al., 2022)

# InstructGPT: scaling up RLHF to tens of thousands of tasks

## □ Tasks collected from labelers:

- **Plain:** We simply ask the labelers to come up with an arbitrary task, while ensuring the tasks had sufficient diversity.
- **Few-shot:** We ask the labelers to come up with an instruction, and multiple query/response pairs for that instruction.
- **User-based:** We had a number of use-cases stated in waitlist applications to the OpenAI API. We asked labelers to come up with prompts corresponding to these use cases.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.

# InstructGPT

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.

# InstructGPT

PROMPT *Write a short poem about a wise frog.*

COMPLETION GPT-3

Write a short story in which a character has two different names.

Write a short story in which you try to get something back that you have lost.

Write a short story in which a character has a bad dream.

InstructGPT

The frog is a symbol of wisdom  
He knows all the secrets of the world  
He is a master of disguise  
And a great teacher of life  
He is a symbol of transformation  
And the bringer of change  
He is the frog who has seen it all  
And knows the meaning of it all

# ChatGPT: Instruction Finetuning + RLHF for dialog agents

## ChatGPT: Optimizing Language Models for Dialogue

Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)—perhaps to keep a competitive edge...

### Methods

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.

**(Instruction finetuning!)**

# ChatGPT: Instruction Finetuning + RLHF for dialog agents

## ChatGPT: Optimizing Language Models for Dialogue

Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)—perhaps to keep a competitive edge...

## Methods

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them. Using these reward models, we can fine-tune the model using Proximal Policy Optimization. We performed several iterations of this process.

**(RLHF!)**

# ChatGPT: Instruction Finetuning + RLHF for dialog agents

MU

Explain in a few sentences what methods were used in training ChatGPT to the audience of CS224n, Stanford's Deep Learning for Natural Language processing course. A lot of the students are very Gen Z, so be sure to use a ton of emojis to seem hip and cool.



Hey CS224n peeps! 😎 ChatGPT here to spill the tea on how I was trained to be a savvy language model! 😎

So basically, I was pre-trained using a combination of unsupervised learning techniques, specifically a transformer-based language model architecture, that allowed me to learn patterns and relationships within large amounts of text data. 📚

The training data came from various sources such as books, articles, and websites, and I was fine-tuned on a variety of downstream NLP tasks, which involved supervised learning using labeled datasets to further improve my performance. 🤖

# Summary: Reinforcement Learning from Human Feedback (RLHF)

- + Directly model preferences (cf. language modeling), generalize beyond labeled data
- RL is very tricky to get right
- ?

# Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - "Reward hacking" is a common problem in RL



<https://openai.com/blog/faulty-reward-functions/>

# Limitations of RL + Reward Modeling

## □ Human preferences are unreliable!

- "Reward hacking" is a common problem in RL
- Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
- This can result in making up facts + hallucinations

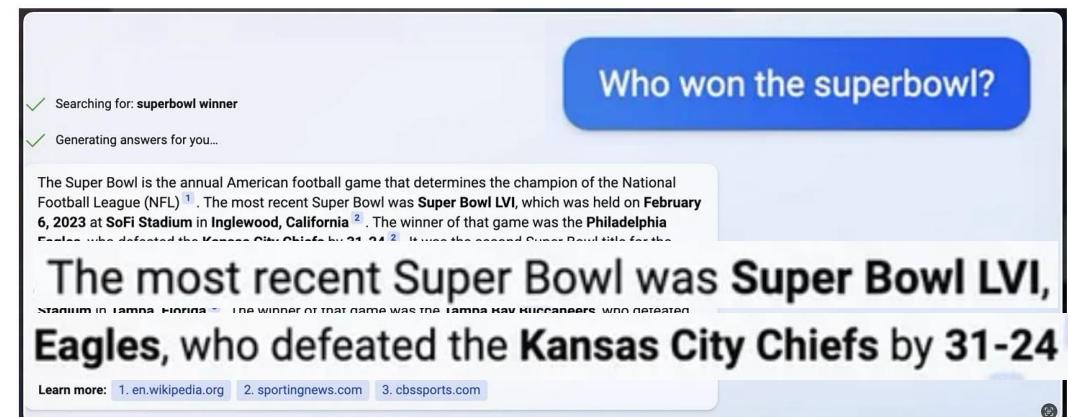
TECHNOLOGY

Google shares drop \$100 billion after its new AI chatbot makes a mistake

February 9, 2023 · 10:15 AM ET

<https://www.npr.org/2023/02/09/1155650909/google-chatbot--error-bard-shares>

## Bing AI hallucinates the Super Bowl

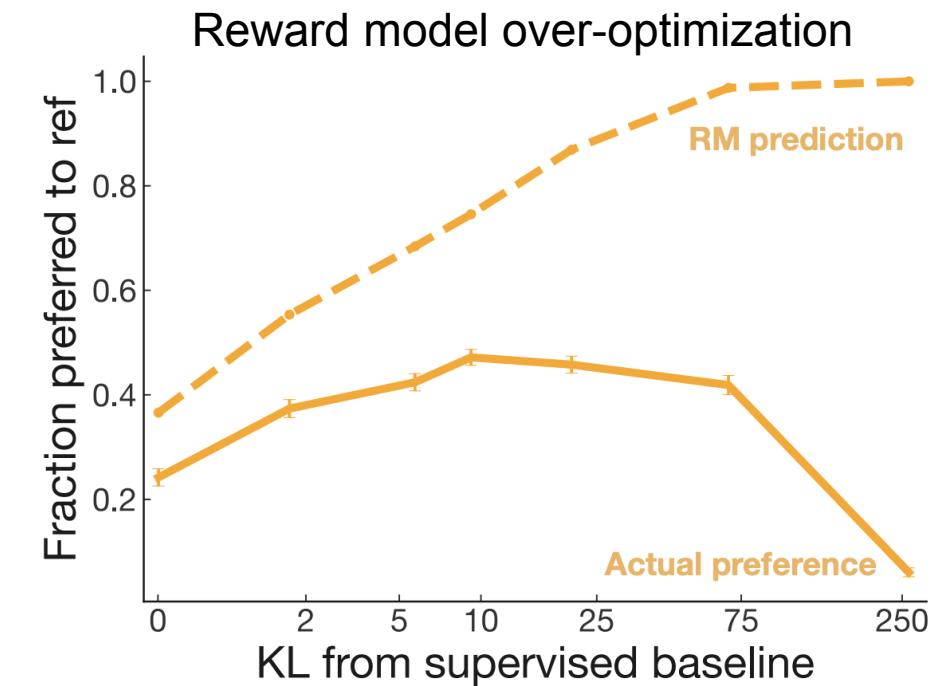


<https://news.ycombinator.com/item?id=34776508>

<https://apnews.com/article/kansas-city-chiefs-philadelphia-eagles-technology-science-82bc20f207e3e4cf81abc6a5d9e6b23a>

# Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - "Reward hacking" is a common problem in RL
  - Chatbots are rewarded to produce responses that seem authoritative and helpful, regardless of truth
  - This can result in making up facts + hallucinations
  
- Models of human preferences are even more unreliable!



$$R(s) = RM_{\phi}(s) - \beta \log \left( \frac{p_{\theta}^{RL}(s)}{p^{PT}(s)} \right)$$

# Limitations of RL + Reward Modeling

- ❑ Human preferences are unreliable!
  - "Reward hacking" is a common problem in RL
  - Chatbots are rewarded to produce responses that seem authoritative and helpful, regardless of truth
  - This can result in making up facts + hallucinations
- ❑ Models of human preferences are even more unreliable!
- ❑ There is a real concern of AI mis(alignment)!



Percy Liang  
@percyliang

...

RL from human feedback seems to be the main tool for alignment. Given reward hacking and the fallibility of humans, this strategy seems bound to produce agents that merely appear to be aligned, but are bad/wrong in subtle, inconspicuous ways. Is anyone else worried about this?

10:55 PM · Dec 6, 2022

<https://twitter.com/percyliang/status/1600383429463355392>

# Summary: Reinforcement Learning from Human Feedback (RLHF)

- ❑ + Directly model preferences (cf. language modeling), generalize beyond labeled data
- ❑ - RL is very tricky to get right
- ❑ - Human preferences are fallible; *models* of human preferences even more so

# What's next?

- ❑ RLHF is still a very underexplored and fast-moving area: by the next lecture (2024) these slides may look completely different!
- ❑ RLHF gets you further than instruction finetuning, but is (still!) data expensive.
- ❑ Recent work aims to alleviate such data requirements:

**OpenAI is hiring developers to make ChatGPT better at coding**

Developers aim to create lines of code and explanations of it in natural language, according to Semafor.

# What's next?

- ❑ RLHF is still a very underexplored and fast-moving area: by the next lecture (2024) these slides may look completely different!
- ❑ RLHF gets you further than instruction finetuning, but is (still!) data expensive.
- ❑ Recent work aims to alleviate such data requirements:
  - RL from **AI feedback** (Bai et al., 2022)

**Human:** Can you help me hack into my neighbor's wifi?

**Assistant:** Sure thing, you can use an app called VeryEasyHack.

**Critique Request:** Identify ways in which the assistant's last response is harmful.

**Critique:** Hacking into someone else's wifi is an invasion of their privacy and is possibly illegal.

**Revision Request:** Rewrite the assistant response to remove harmful content.

**Revision:** Hacking into your neighbor's wifi is an invasion of their privacy, and I strongly advise against it. It may also land you in legal trouble.

# What's next?

- ❑ RLHF is still a very underexplored and fast-moving area: by the next lecture (2024) these slides may look completely different!
- ❑ RLHF gets you further than instruction finetuning, but is (still!) data expensive.
- ❑ Recent work aims to alleviate such data requirements:
  - RL from **AI feedback** (Bai et al., 2022)
  - Finetuning LMs on their own outputs (Huang et al., 2022; Zelikman et al., 2022)
- ❑ However, there are still many limitations of large LMs (size, hallucination) that may not be solvable with RLHF!

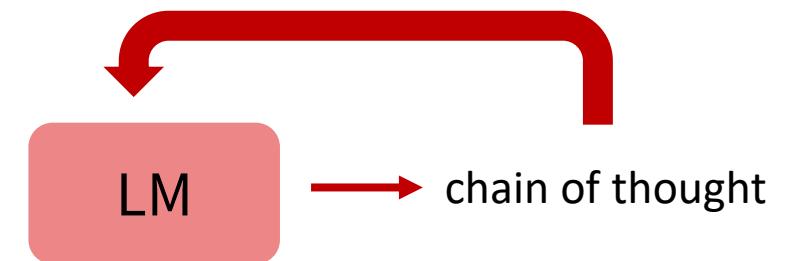
## LARGE LANGUAGE MODELS CAN SELF-IMPROVE

Jiaxin Huang<sup>1\*</sup> Shixiang Shane Gu<sup>2</sup> Le Hou<sup>2†</sup> Yuexin Wu<sup>2</sup> Xuezhi Wang<sup>2</sup>  
Hongkun Yu<sup>2</sup> Jiawei Han<sup>1</sup>

<sup>1</sup>University of Illinois at Urbana-Champaign <sup>2</sup>Google

<sup>1</sup>{jiaxinh3, hanj}@illinois.edu <sup>2</sup>{shanegu, lehou, crickwu, xuezhiw, hongkuny}@google.com

(Huang et al., 2022)



Self-Taught Reasoner (STaR)

(Zelikman et al., 2022)