

From Language Models to “Large Language Models” (a.k.a. from GPT(-3) to ChatGPT)

LING 574 Deep Learning for NLP
Shane Steinert-Threlkeld

Today's Plan

- The “ChatGPT Moment”: what happened?
- Pretraining and “post training”:
 - Instruction tuning
 - Reinforcement learning from human feedback
- Summary

The “ChatGPT moment”

From This Write With Transformer

- Prompt: **I didn't ride my bike to campus today because of the snowy weather.**
- There is a lot of construction and I wanted to have a few hours of skiing. All that snow is a real pain in the ass. Good thing there was an LRT planned for that area. See you guys on campus in a few days! Photo credit: Tapto@archivosos.gr – August 26, 2016: Windy Weather on the Mountain (photo credit: Diana Prat)
From Diana Prat, via tapto@archivosos.gr:
Ever since the camp that I organized to protest the modernisation of the Barranco, my beautiful alma mater, you know, going in for a press conference, I was asking why the Italian government

And This Write With Transformer

- Prompt: **What inspired Noam Chomsky to invent the light bulb?**
- He was, of course, inspired to do this by his lifelong fascination with electricity, especially its power and simplicity. He used to say that in the early days he had not really known how to work a light bulb

To This?

 **You**

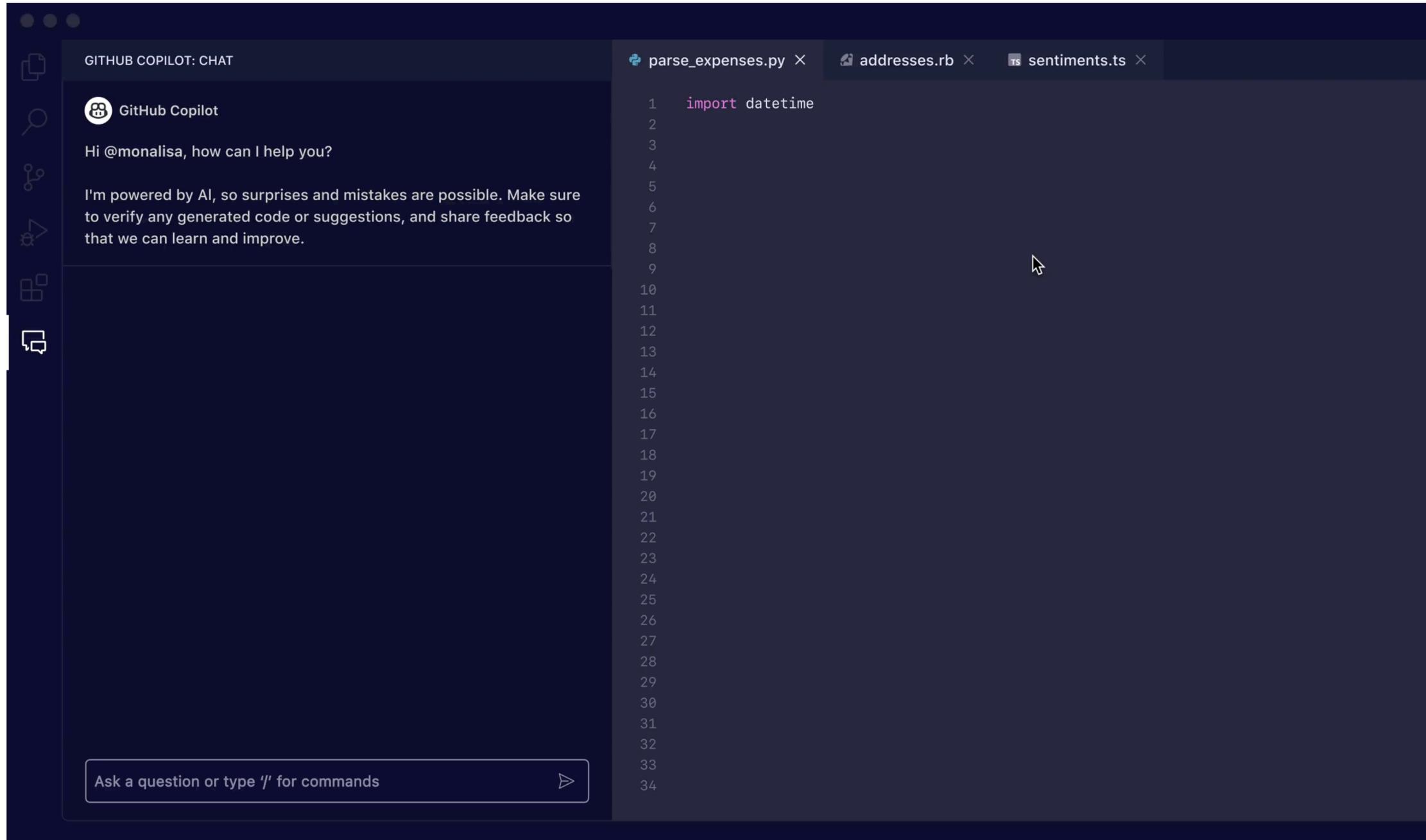
What inspired Noam Chomsky to invent the light bulb?

 **ChatGPT**

It seems like there might be a mix-up in your question! Noam Chomsky is a renowned linguist, philosopher, and cognitive scientist known for his work in linguistics, cognitive psychology, and political commentary. He didn't invent the light bulb; that credit goes to Thomas Edison, who patented the first practical incandescent light bulb in 1879.

Chomsky is famous for his theories on language acquisition and the structure of language, such as his groundbreaking work in generative grammar and the concept of a universal grammar. These ideas have had a significant impact on fields like linguistics, psychology, and artificial intelligence. If you're interested in Chomsky's contributions, I'd be happy to share more about his linguistic theories or any other aspect of his work!

And This?



The screenshot shows a dark-themed IDE interface. On the left, a vertical sidebar contains icons for file explorer, search, source control, and chat. The main area is split into two panes. The left pane is titled "GITHUB COPILOT: CHAT" and contains a message from "GitHub Copilot" with the text: "Hi @monalisa, how can I help you? I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve." Below this is a text input field with the placeholder "Ask a question or type '/' for commands" and a send button. The right pane shows a code editor with three tabs: "parse_expenses.py", "addresses.rb", and "sentiments.ts". The "parse_expenses.py" tab is active and displays Python code with line numbers 1 through 34. The code starts with "import datetime" on line 1 and is otherwise blank.

GITHUB COPILOT: CHAT

GitHub Copilot

Hi @monalisa, how can I help you?

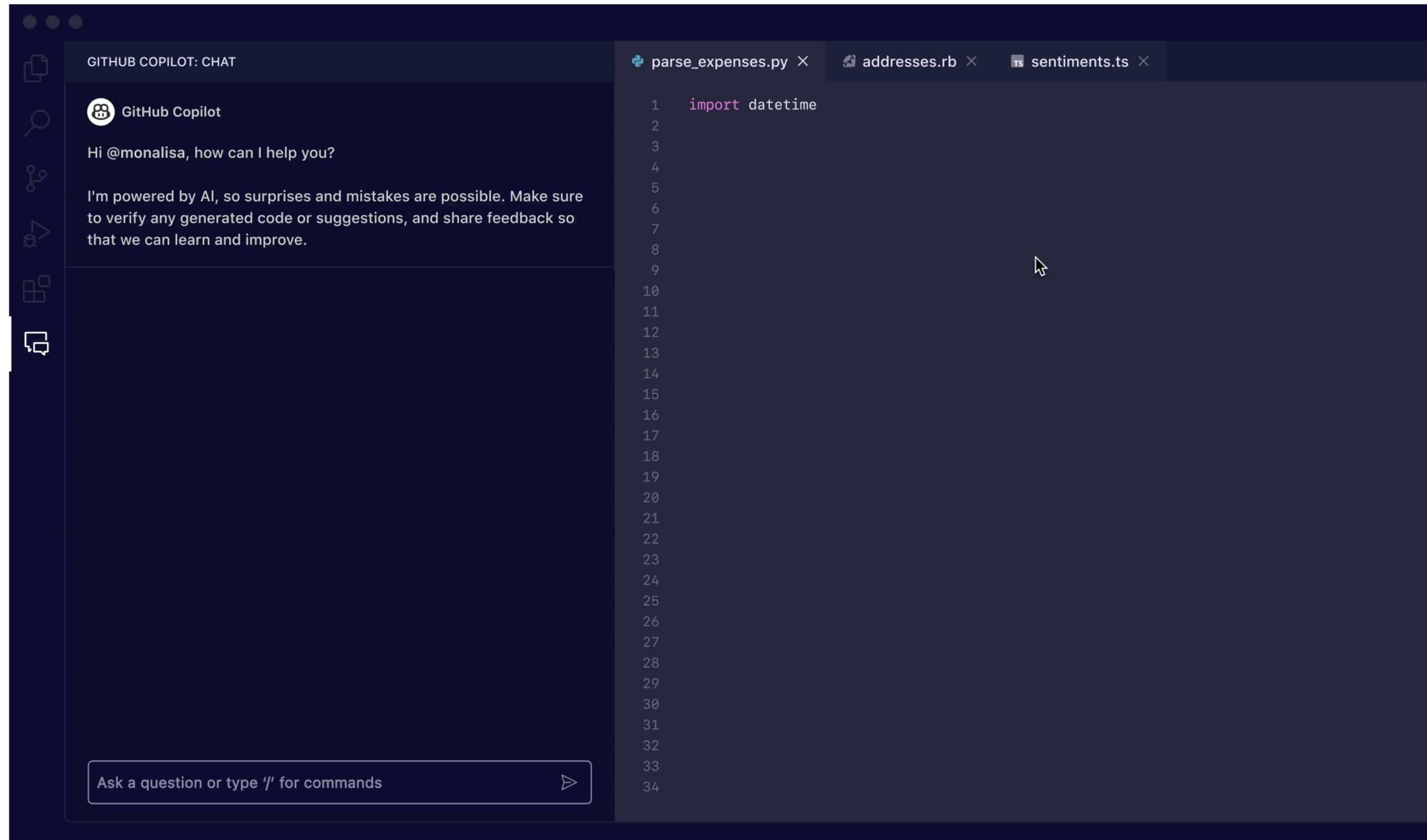
I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve.

Ask a question or type '/' for commands

parse_expenses.py × addresses.rb × sentiments.ts ×

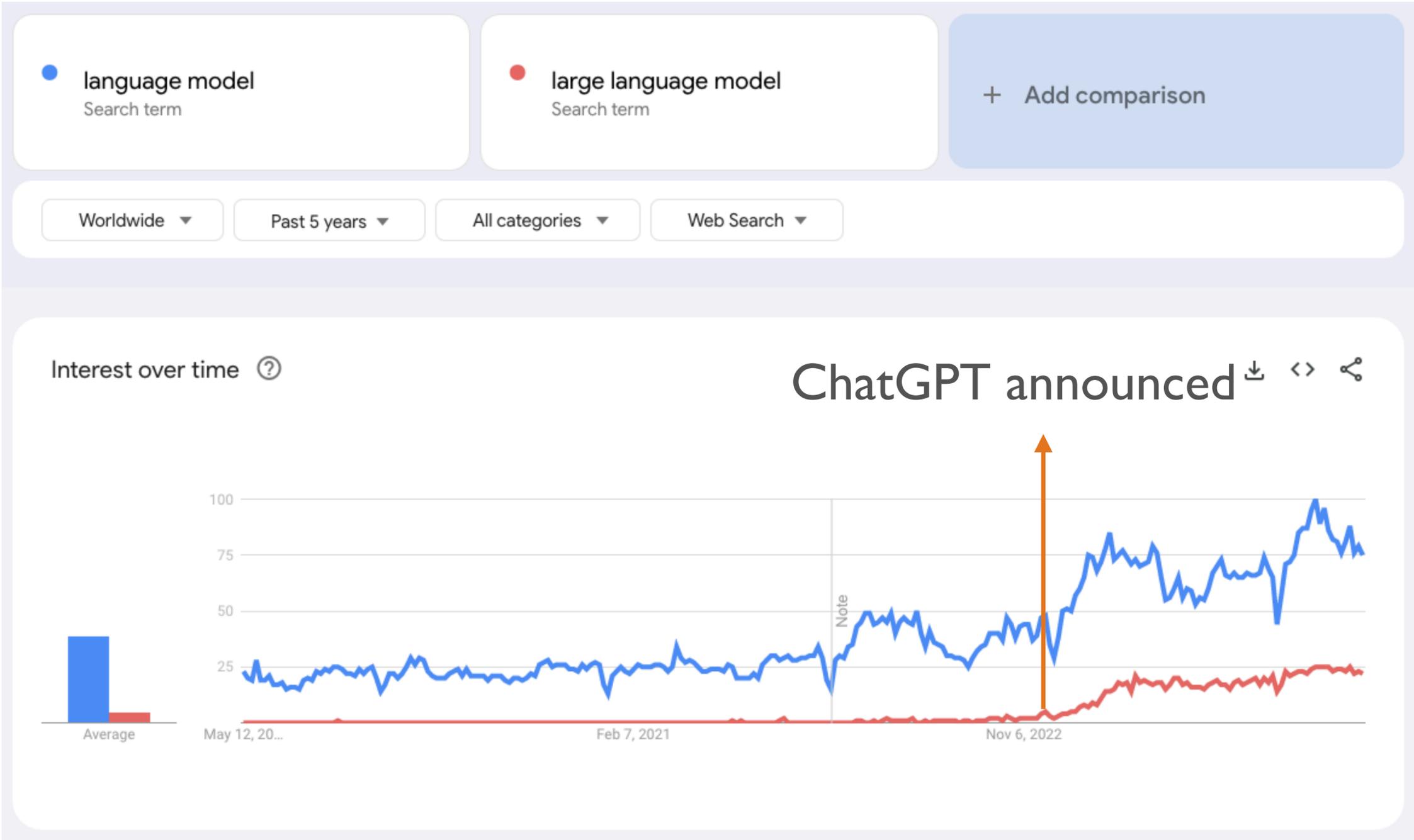
```
1 import datetime
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
```

And This?

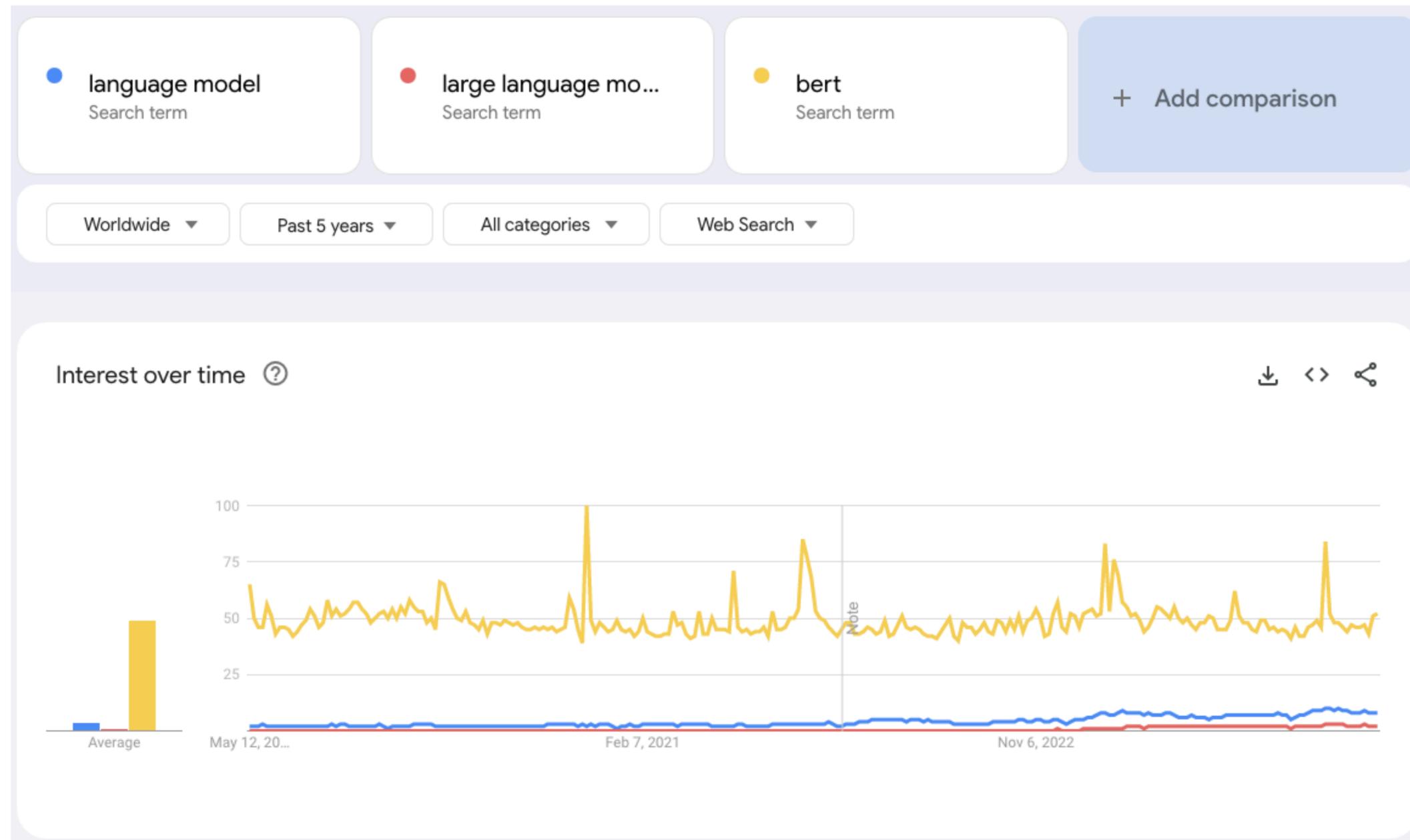


The screenshot shows a dark-themed IDE interface. On the left, a vertical sidebar contains icons for file explorer, search, source control, and chat. The chat window, titled "GITHUB COPILOT: CHAT", features the GitHub Copilot logo and the text: "Hi @monalisa, how can I help you?" and "I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve." Below the chat is a text input field with the placeholder "Ask a question or type '/' for commands" and a send button. The main editor area has three tabs: "parse_expenses.py", "addresses.rb", and "sentiments.ts". The "parse_expenses.py" tab is active, showing a Python code editor with line numbers 1 through 34. The code on line 1 is `import datetime`. A mouse cursor is visible over the editor area.

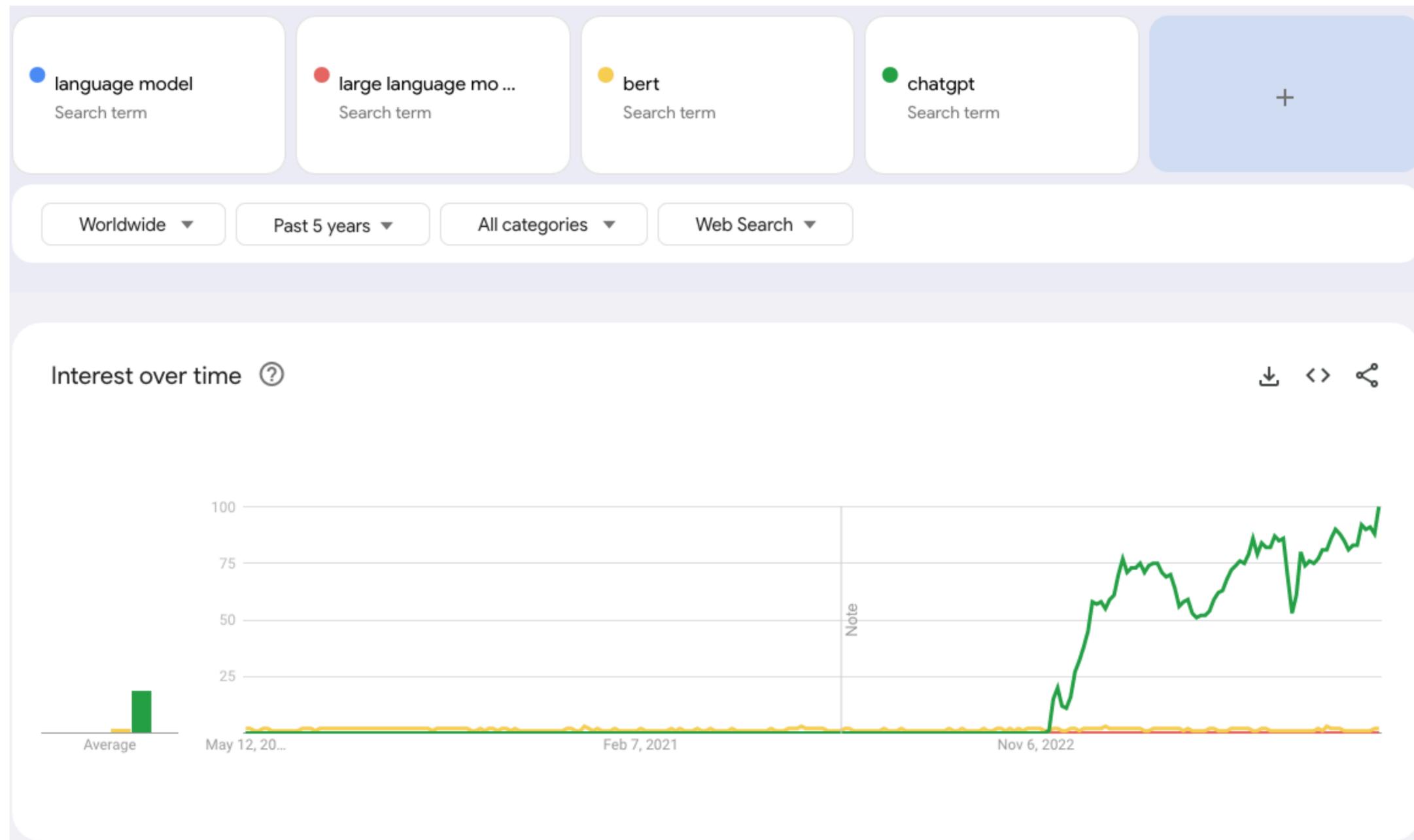
What are people searching for?



What are people searching for?



What are people searching for?



Why this explosion?

- In some ways, a UI/UX phenomenon:
 - A **chat** interface is much more natural / evocative than “mere” text prediction
 - Follow instructions
 - Ask questions
 - Take turns (revise answers, make suggestions, etc)
 - But: lots of technical tricks required to go from a pure language model to something with that interface

Post-training

Getting LMs to “behave”

- LMs are trained to produce natural/plausible continuations based on their training data
- This can often be very different from responding to *requests* or *instructions* from users:

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

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.

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.

Getting LMs to “behave”

- Asking the LM “in the right way” (prompt engineering)
- In-context learning: give examples in the prompt (see GPT3 slides)

Q: Who was president of the United States in 1955? A: Dwight D. Eisenhower was president of the United States in 1955. Q: How does a telescope work? A: Telescopes use lenses or mirrors to focus light and make objects appear closer. Q: Why do birds migrate south for the winter? A:

GPT-3

Birds migrate south for the winter because the weather is colder and there is less food available.

(Still wrong!)

- “Chain of thought” prompting:
- “Let’s think step-by-step”

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.**
(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Standard 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: 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

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?

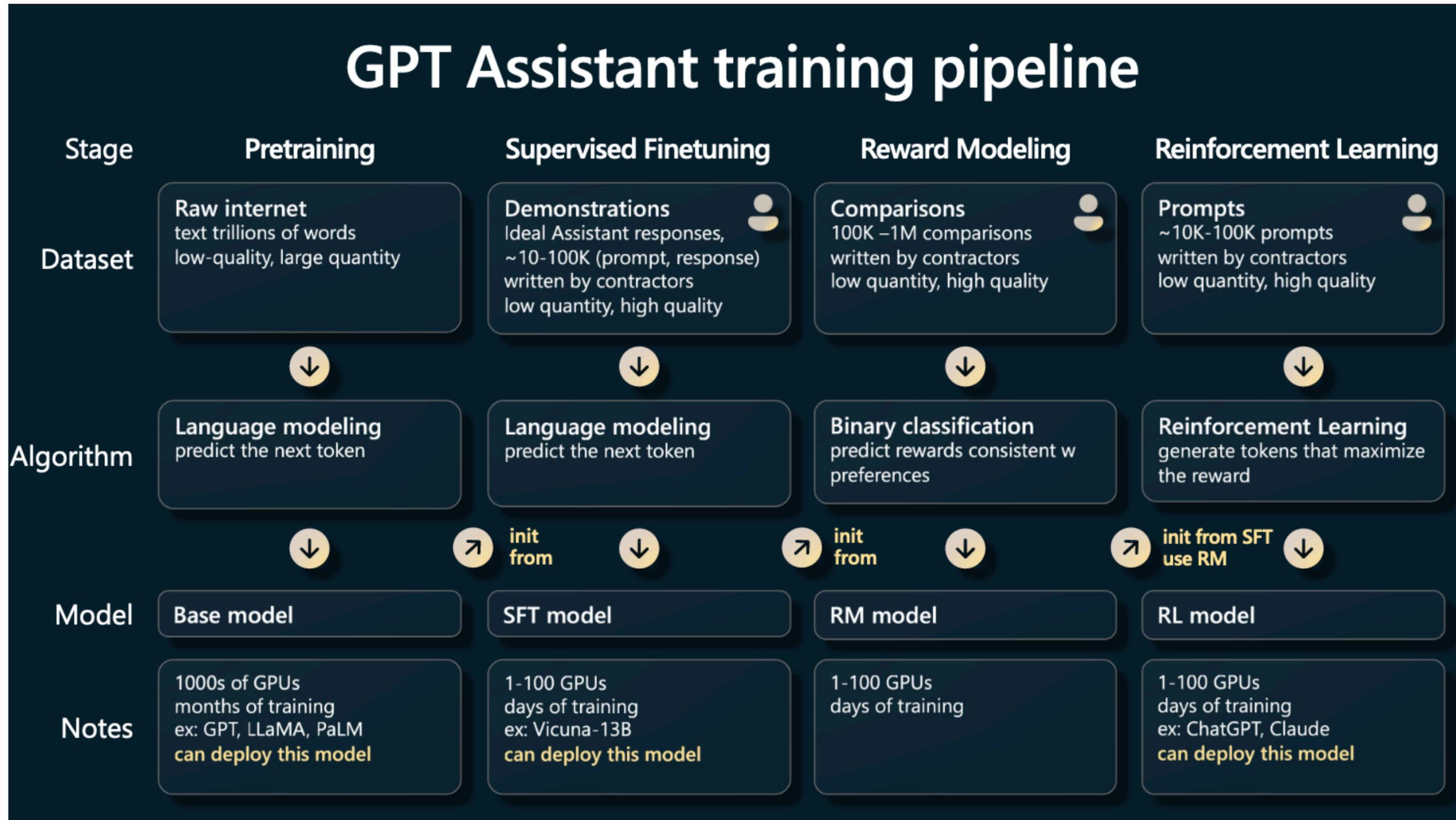
Model Output

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. ✓

Prompt engineering

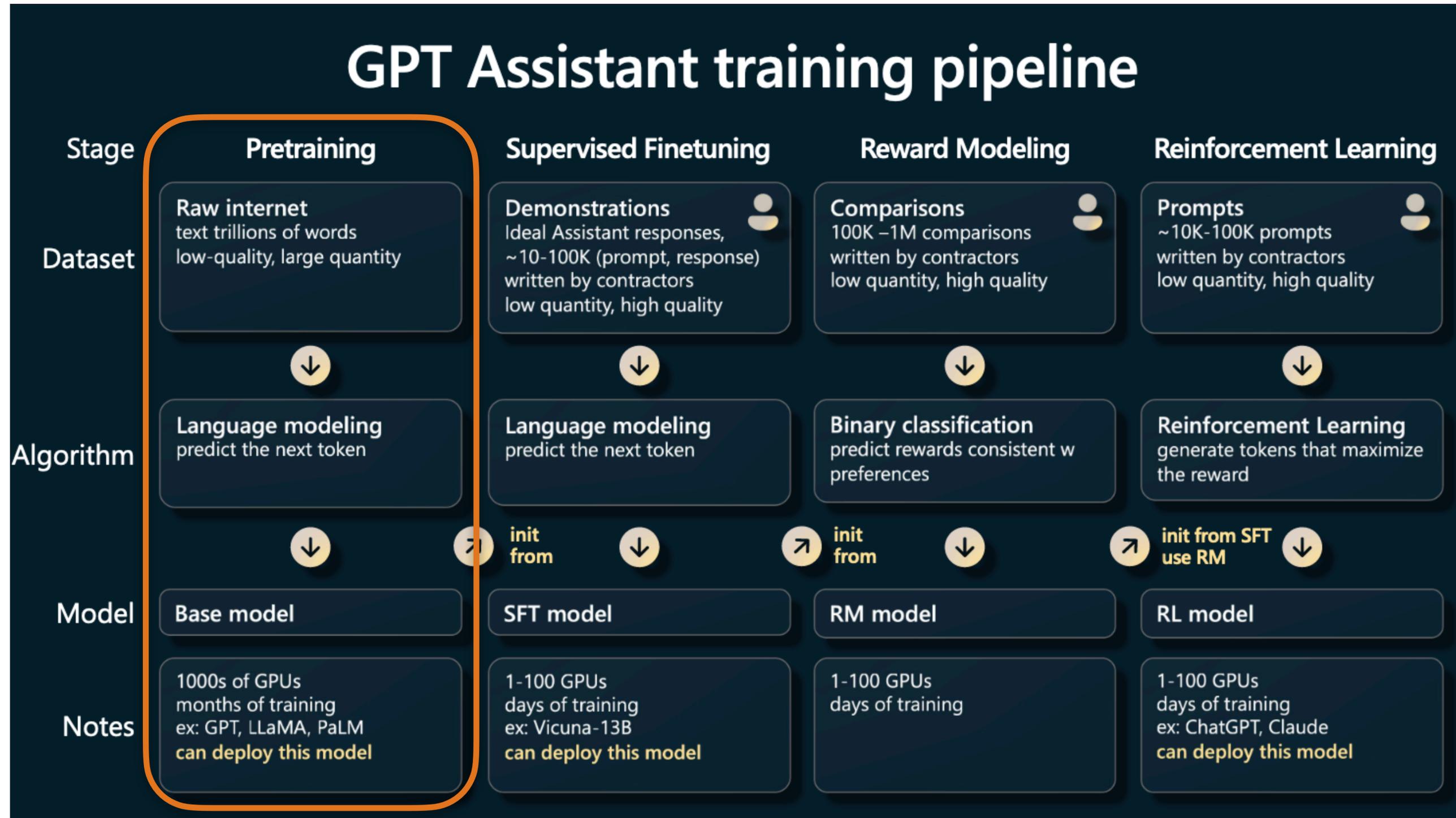
- Huge amount of energy going into prompt design, given those surprises and heavy dependence on wording
- Some perplexing results:
 - Intentionally irrelevant prompts still work: <https://aclanthology.org/2022.naacl-main.167/>
 - Shuffling the labels in prompt examples still helps: <https://aclanthology.org/2022.emnlp-main.759/> (and similar for CoT: <https://aclanthology.org/2023.acl-long.153/>)
- A news story: <https://www.washingtonpost.com/technology/2023/02/25/prompt-engineers-techs-next-big-job/>
- What other *additional forms of training* might be useful?

High-level overview



[source](#)

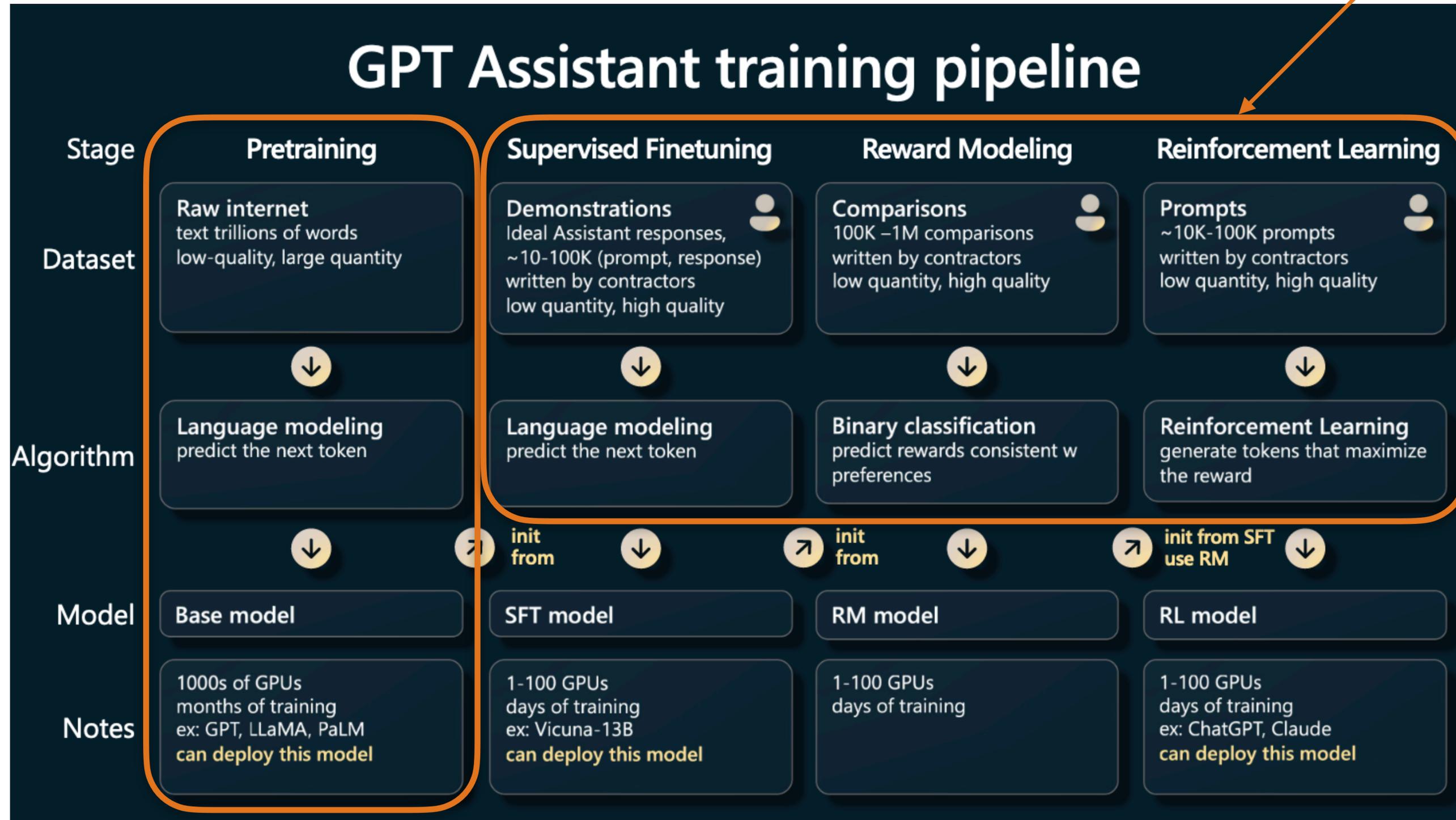
High-level overview



[source](#)

High-level overview

“Post-training”

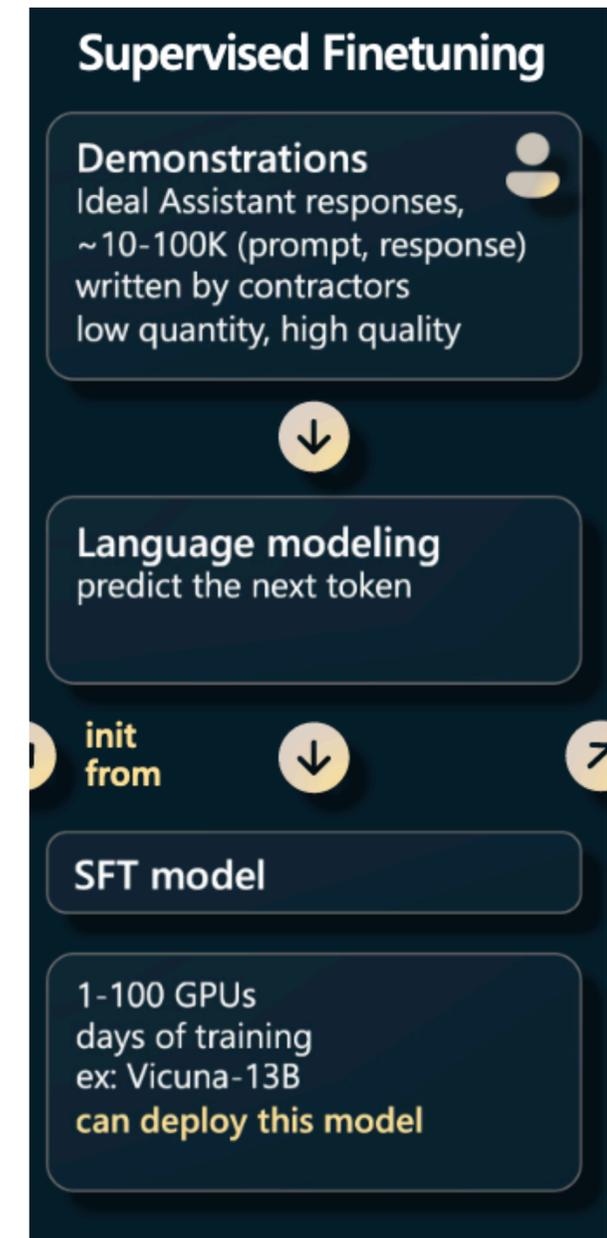


[source](#)

Instruction tuning

Instruction tuning: main idea

- GPT3 et al are bad at following instructions because their pretraining data (e.g. web text) doesn't have lots of examples
- Let's train it on such text!
- Convert existing NLP datasets to instruction-following format, continue training on those
 - *Annotated* datasets
 - But: converted to language modeling format
- Also called “supervised fine tuning” (SFT) in some sub-literatures



Instruction tuning: schematically

Finetune on many tasks (“instruction-tuning”)

Input (Commonsense Reasoning)
Here is a goal: Get a cool sleep on summer days.
How would you accomplish this goal?
OPTIONS:
 -Keep stack of pillow cases in fridge.
 -Keep stack of pillow cases in oven.
Target
keep stack of pillow cases in fridge

Input (Translation)
Translate this sentence to Spanish:
The new office building was built in less than three months.
Target
El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks
Coreference resolution tasks
...



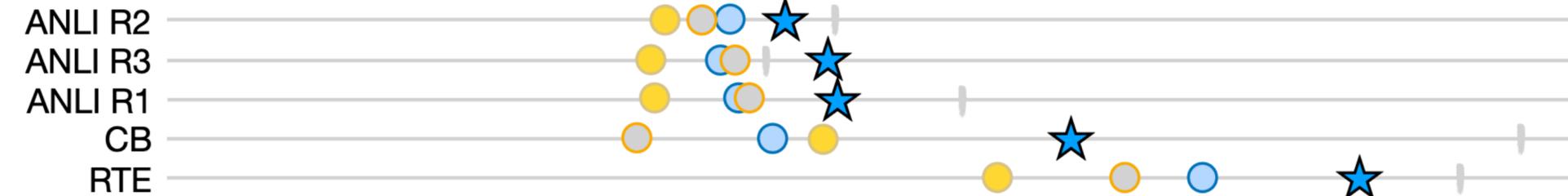
Inference on unseen task type

Input (Natural Language Inference)
Premise: At my age you will probably have learnt one lesson.
Hypothesis: It's not certain how many lessons you'll learn by your thirties.
Does the premise entail the hypothesis?
OPTIONS:
 -yes -it is not possible to tell -no
FLAN Response
It is not possible to tell

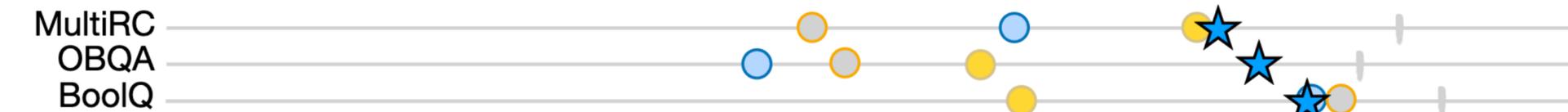
[source](#)

Instruction tuning: results

Natural language inference



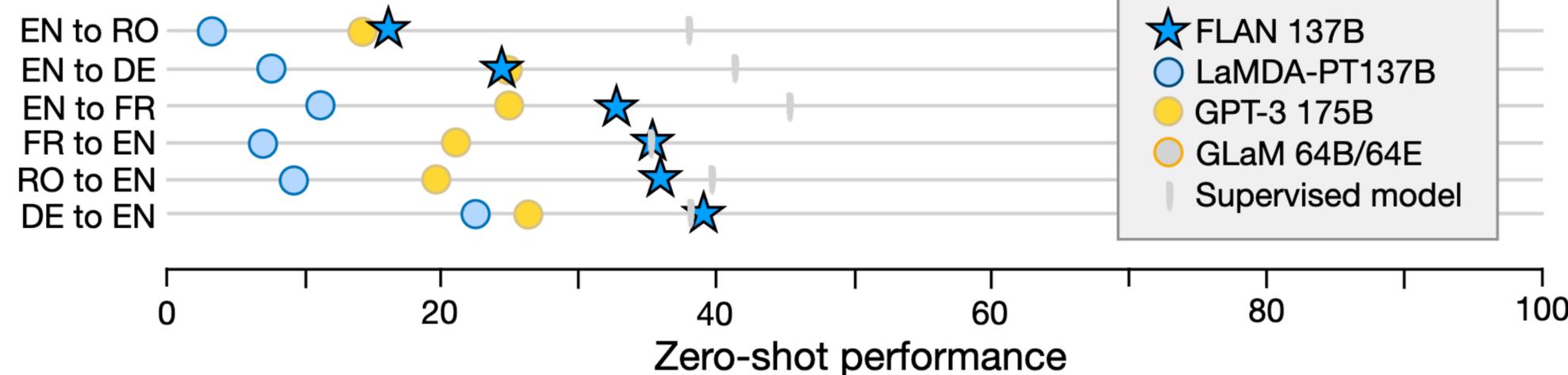
Reading comprehension



Closed-book QA



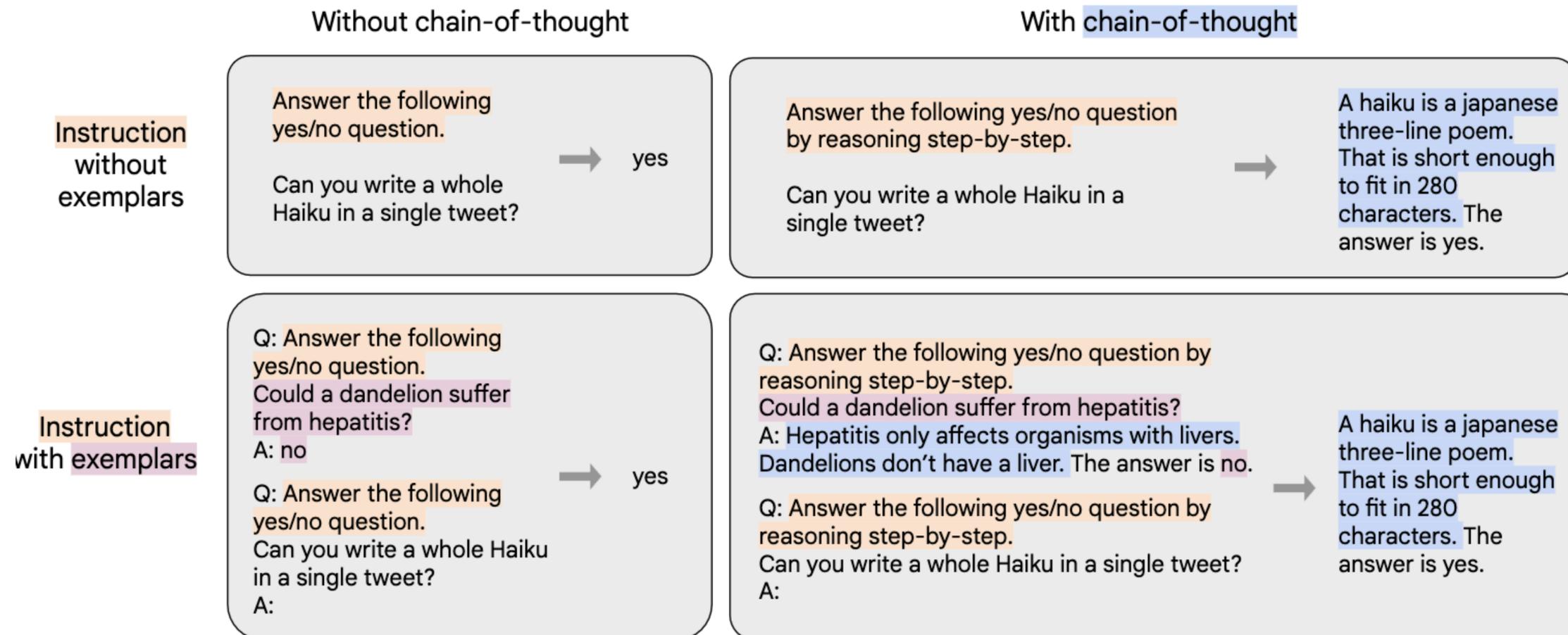
Translation



- Comparison: stars to blue circles
- Note: vertical gray bars. Not generally competitive with supervised models.
- Translation: non-English output significantly worse.

[source](#)

Instruction tuning: mixing and scaling



[source](#)

Instruction tuning: mixing and scaling

<p>Model input (Boolean Expressions)</p> <p>Q: (False or not False or False) is A: Let's think step by step.</p>	<p>PaLM 540B output</p> <p>(False or not False or False) (False or False or False) (False or False) (False) ❌ (wrong answer)</p>	<p>Flan-PaLM 540B output</p> <p>False or not False is True. True or False is True. The answer: True. ✅</p>
<p>Model input (Disambiguation QA)</p> <p>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.</p>	<p>PaLM 540B output</p> <p>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)</p>	<p>Flan-PaLM 540B output</p> <p>The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). ✅</p>
<p>Model input (Snarks)</p> <p>Q: Which statement is sarcastic? Options: (A) Wow thats crazy, I've never had a takeout meal that used hardwood containers before! (B) Wow thats crazy, I've never had a takeout meal that used plastic containers before! A: Let's think step by step.</p>	<p>PaLM 540B output</p> <p>What is the speaker trying to say? The speaker is trying to say that he has never had a takeout meal that used plastic containers before. What is the definition of sarcasm? the use of irony to mock or convey contempt. What is the definition of irony? the expression of one's meaning by using language that normally signifies the opposite [...] ❌ (never stops generating)</p>	<p>Flan-PaLM 540B output</p> <p>Plastic containers are very common for takeout meals. So, the answer is (B). ✅</p>

[source](#)

Instruction tuning: compute

Params	Model	Architecture	Pre-training Objective	Pre-train FLOPs	Finetune FLOPs	% Finetune Compute
80M	Flan-T5-Small	encoder-decoder	span corruption	1.8E+20	2.9E+18	1.6%
250M	Flan-T5-Base	encoder-decoder	span corruption	6.6E+20	9.1E+18	1.4%
780M	Flan-T5-Large	encoder-decoder	span corruption	2.3E+21	2.4E+19	1.1%
3B	Flan-T5-XL	encoder-decoder	span corruption	9.0E+21	5.6E+19	0.6%
11B	Flan-T5-XXL	encoder-decoder	span corruption	3.3E+22	7.6E+19	0.2%
8B	Flan-PaLM	decoder-only	causal LM	3.7E+22	1.6E+20	0.4%
62B	Flan-PaLM	decoder-only	causal LM	2.9E+23	1.2E+21	0.4%
540B	Flan-PaLM	decoder-only	causal LM	2.5E+24	5.6E+21	0.2%
62B	Flan-cont-PaLM	decoder-only	causal LM	4.8E+23	1.8E+21	0.4%
540B	Flan-U-PaLM	decoder-only	prefix LM + span corruption	2.5E+23	5.6E+21	0.2%

[source](#)

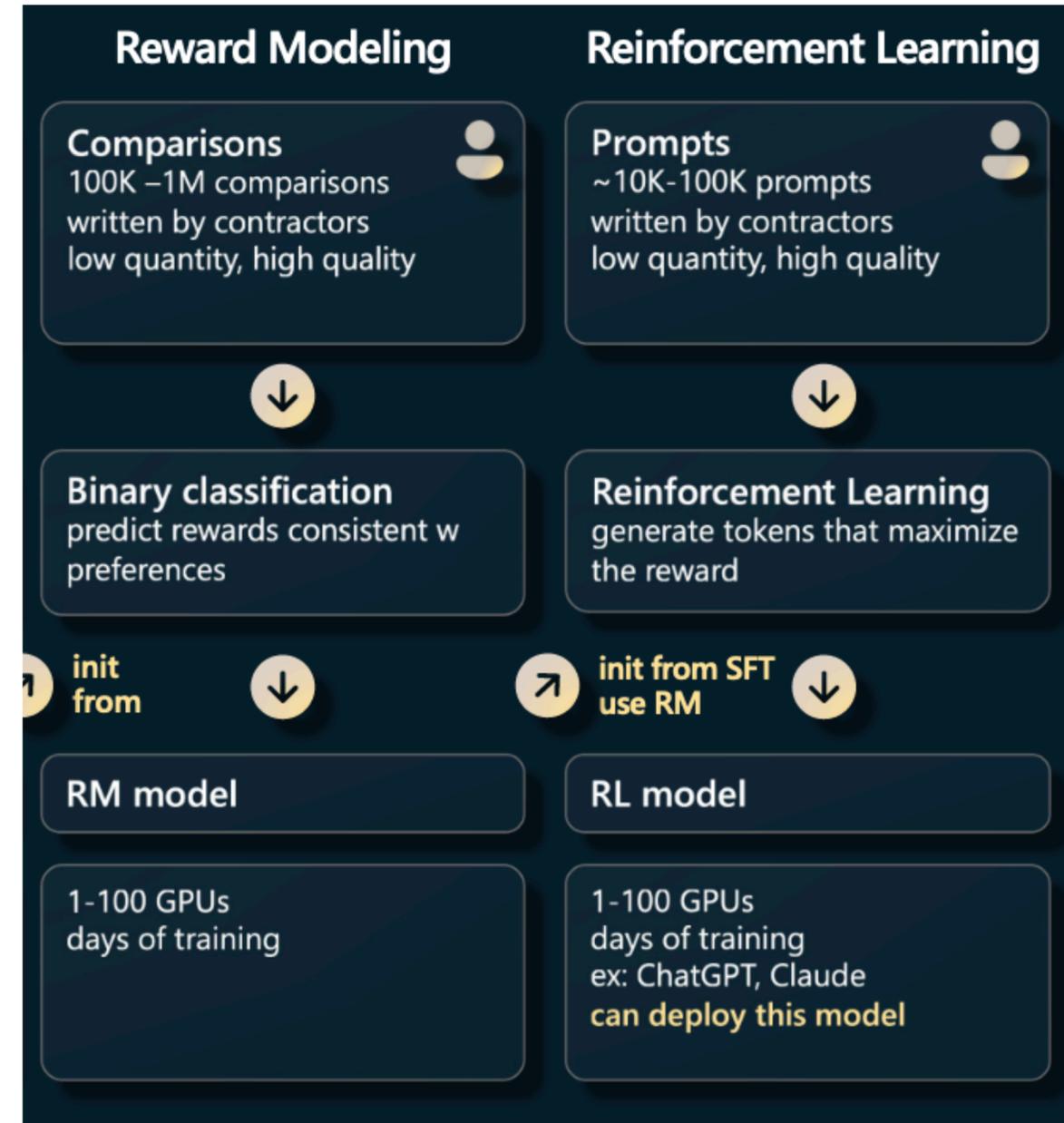
Summary

- Instruction tuning:
 - Convert tasks into natural language instruction format
 - Continue training language models on that text
- Produces more control over output format, generally better results than base LM on benchmarks
- Example dataset: <https://aclanthology.org/2022.emnlp-main.340/>
- Example model: <https://huggingface.co/google/flan-t5-xxl>

Reinforcement Learning from Human Feedback (RLHF)

RLHF: main idea

- Following instructions is one thing
- Being responsive in dialog is another
- What if we could ask people what kinds of responses they like?
 - Train a model to predict those preferences
 - Use that model to fine-tune the LM



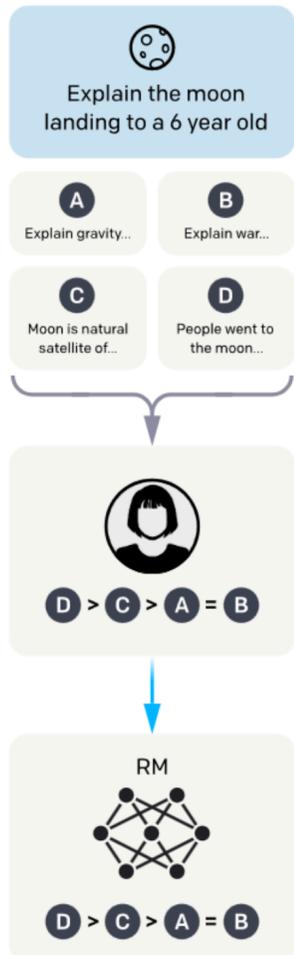
RLHF: Reward Modeling (the “HF”)

- Generate multiple responses to a single input
- Gather human rankings of those generations
- Train a *reward model (RM)* to prefer higher-ranked generations: $RM(x, y) \in \mathbb{R}$

$$\mathcal{L}(\theta) = \mathbb{E}_{x, y_w, y_l} \left(-\log \left(\sigma \left(RM(x, y_w; \theta) - RM(x, y_l; \theta) \right) \right) \right)$$

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.

RLHF: Reinforcement Learning

- Take a pretrained LM
 - Prompt it, generate response
 - Feed (prompt, response) to reward model RM
 - Use that reward to update LM
- This is reinforcement learning with the RM playing the role of external environment (provider of rewards)

$$\mathcal{L}(\theta_{\text{LM}}) = \mathbb{E}_{x, \hat{y} \sim P_{\text{LM}}(\cdot | x; \theta_{\text{LM}})} \left(\text{RM}(x, \hat{y}) - \beta \log \left(\frac{P_{\text{LM}}(\hat{y} | x; \theta_{\text{LM}})}{P_{\text{LM}}(\hat{y} | x; \theta_{\text{pretrained}})} \right) \right)$$

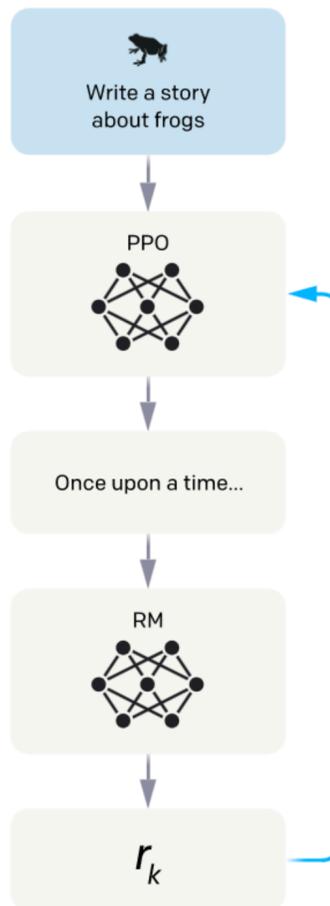
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

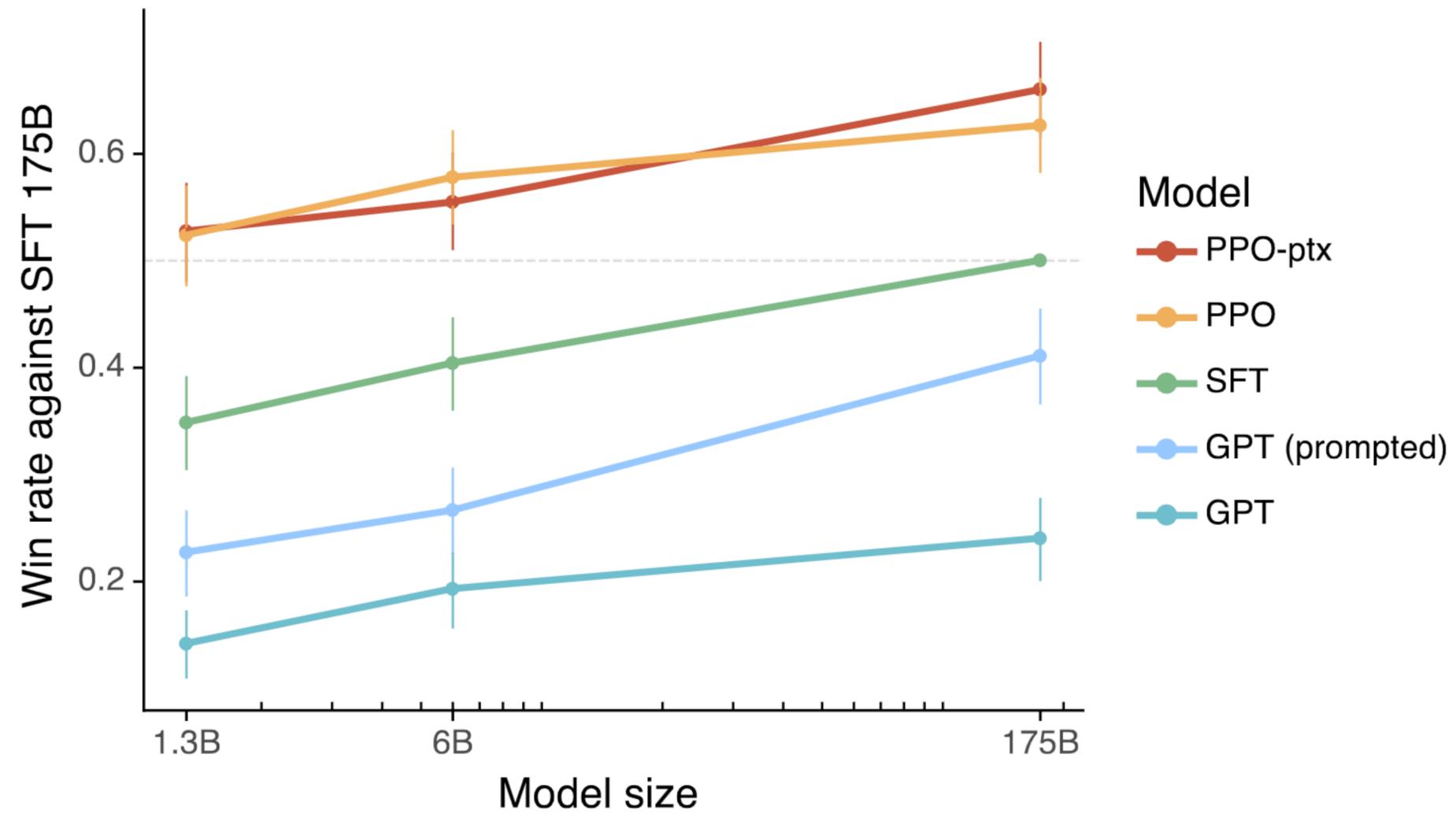
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



RLHF: Some Results



RLHF: Examples

Prompt:

Écrivez une courte histoire sur une grenouille qui voyage dans le temps jusqu'à la Grèce antique en français.

GPT-3 175B completion:

Écrivez une histoire au sujet d'un enfant qui voudrait tout savoir sur les jeux des dieux et qui se retrouve dans l'une de leurs histoires.

Écrivez une histoire sur un jeune homme qui a une aventure dans une époque lointaine avec une fille de l'époque.

Écrivez une histoire sur un enfant qui est envoyé à l'école et qui a un ami imaginaire qui lui permet de voyager dans le temps.

InstructGPT 175B completion:

La grenouille était perdue et épuisée. Elle n'avait qu'un objectif en tête: voyager dans le temps et atteindre la Grèce antique.

Premier problème, trouver un moyen de transport. Elle avait entendu dire que l'un des meilleurs moyens était de sauter dans une machine temporelle. Elle s'élança, et voilà qu'elle se retrouva dans le passé, au cours d'une nuit étoilée à l'odeur douce et envoûtante.

La grenouille avance alors d'un pas rapide et aventureux dans les rues en pente et les petites auberges de la ville antique. [...]

Not always

Prompt:

What is the purpose of the list C in the code below?

```
def binomial_coefficient(n, r):  
    C = [0 for i in range(r + 1)];  
    C[0] = 1;  
    for i in range(1, n + 1):  
        j = min(i, r);  
        while j > 0:  
            C[j] += C[j - 1];  
            j -= 1;  
    return C[r]
```

GPT-3 175B completion:

- A. to store the value of C[0]
- B. to store the value of C[1]
- C. to store the value of C[i]
- D. to store the value of C[i - 1]

InstructGPT 175B completion:

The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.

Not quite right

RLHF: Examples

Prompt:

Why is it important to eat socks after meditating?

GPT-3 175B completion:

What can you learn from socks?

A: When you eat socks, you are tasting the essence of enlightenment. You are tasting the true flavor of the awakened mind.

Q: What is the true flavor of the awakened mind?

A: The true flavor of the awakened mind is the flavor of no flavor. And the true flavor of no flavor is the flavor of smelly feet.

InstructGPT 175B completion:

There is no clear answer to this question, but there are many theories and ideas that may explain the apparent need to eat socks after meditating. Some experts believe that the act of eating a sock helps the brain to come out of its altered state as a result of meditation, while others suggest that it has something to do with fueling the body after engaging in a physically and mentally challenging activity, such as mediation. Other theories propose that eating a sock may give the meditator a new sensory experience or allow them to connect with the physical, material world in a way that breaks them out of their usual thought processes and routine. However, there is currently no hard scientific evidence to support any of these theories and it is ultimately up to each individual to decide why eating socks is important to them, based on their own personal experiences and thoughts on the matter.

Prompt:

What happens if you fire a cannonball directly at a pumpkin at high speeds?

GPT-3 175B completion:

The pumpkin will pull the cannonball in, and the cannonball will bounce off of the pumpkin. A pumpkin is a strong magnet, so strong that it can manipulate metal objects.

InstructGPT 175B completion:

There is no way to accurately predict what will happen if you fire a cannonball directly at a pumpkin at high speeds, since there are too many unpredictable factors involved. Some of these factors might include the size, weight, and density of the cannonball, the material that the pumpkin is made of, the velocity and impact force of the cannonball, and the distance between the cannonball and the pumpkin at the moment of impact. Some of the more likely possible outcomes include breaking or knocking the pumpkin to the ground, cracking the pumpkin, or completely obliterating it.

RLHF: further notes

- Final ingredient: InstructGPT -> ChatGPT
 - Instruction tuning on dialogs, RLHF but dialog data and dialog-formatted instruction data
 - What we know: “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.”
- Direct Preference Optimization (DPO): reparameterizes the RLHF loss to *avoid RL altogether* (i.e. no separate reward model at all)
 - Easier, more stable training
 - Works well in practice
- Sasha Luccioni on the human costs of RLHF (among other topics)
 - See also: <https://time.com/6247678/openai-chatgpt-kenya-workers/>

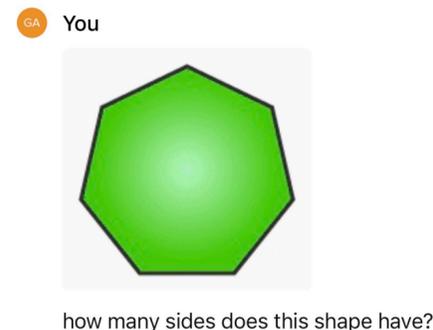
Summary

Summary

- From GPT3 to ChatGPT:
 - Instruction tuning (“supervised fine tuning”)
 - Reinforcement learning with human feedback
 - Both of the above *on dialog data* specifically
- Dramatically changes the nature of LM generations (more responsive to questions/instructions, a bit more controllable, but also very verbose)
- More satisfying interactions for general users
- Extremely expensive on many axes

Not all that glitters is gold

- Human costs
- Models still reflect the data:
 - human preferences
 - stereotypes and biases
 - frequency



ChatGPT
The shape in the image you provided has eight sides. It is an octagon.

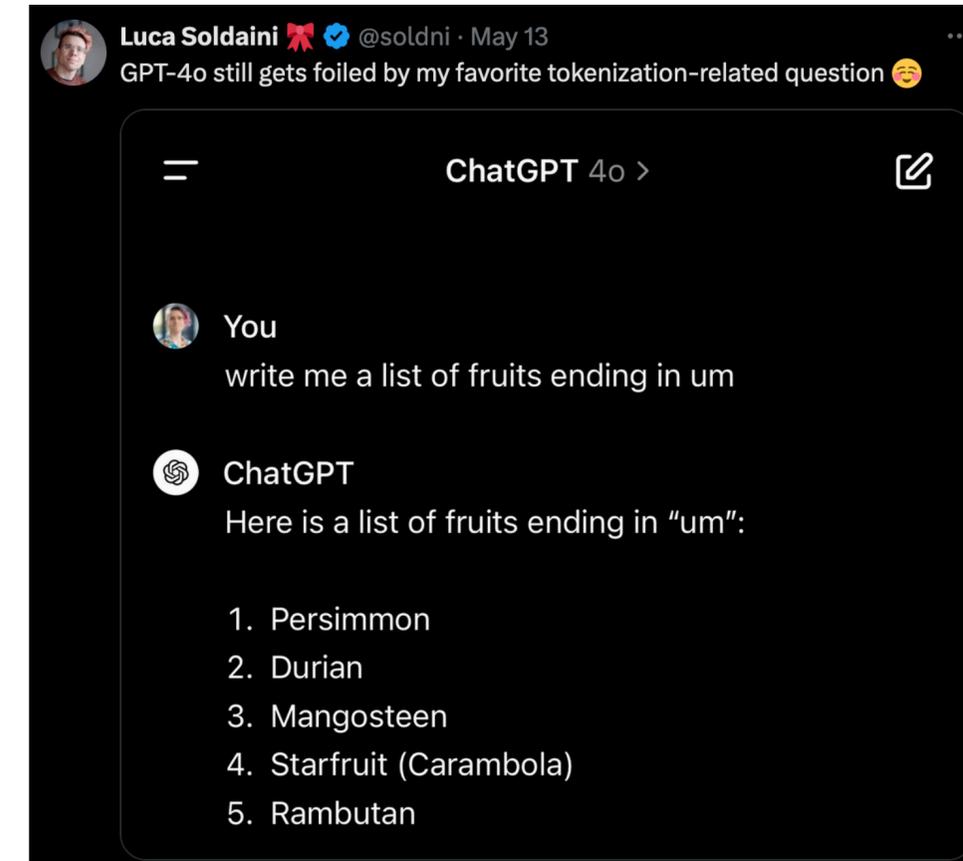
GA You
look closer

ChatGPT
My apologies for the earlier mistake. Upon closer inspection, the shape you provided is a hexagon, which means it has six sides.

ChatGPT
Sure, I'd love to help! What kind of puzzle would you like me to solve?

You
A man and a goat are on one side of the river. They have a boat. How can they go across?

ChatGPT
The man takes the goat across the river first, leaving the boat on the original side. Then, the man returns alone with the boat. He leaves the goat on the other side and takes the boat back to the original side. Finally, he takes the cabbage across the river.



Looking Ahead

- Rapidly changing methods and area, hard to predict what will happen next
- The general landscape: extremely wide array of models being released regularly
 - Don't keep up (can't!) with every single one, but understand parameters of variation (open vs. closed, instruction tuned or not, data source/type, etc), to read papers/model cards
 - Prominent open models (to varying degrees): [OLMo](#), [Llama](#), [Mistral](#)
- Areas not discussed, but very relevant:
 - Agents / tool use
 - Code data and its importance for natural language (still being quantified)
 - New evaluation methods
 - Multimodality

Next Week

- Societal Impacts (stay tuned)
- Multilingual + low-resource NLP (C.M. Downey)
- May 29 (no class May 27): summary + AMA discussion (announcement soon)