

How does ChatGPT work?



Li Chen

University of Louisiana at Lafayette

Representing words

- ❖ Traditional NLP: regard words as **discrete** symbols
- ❖ Such symbols can be represented as **one-hot** vectors:

`motel = [0 0 0 0 0 0 0 0 0 1 0 0 0 0]`

`hotel = [0 0 0 0 0 0 1 0 0 0 0 0 0 0]`

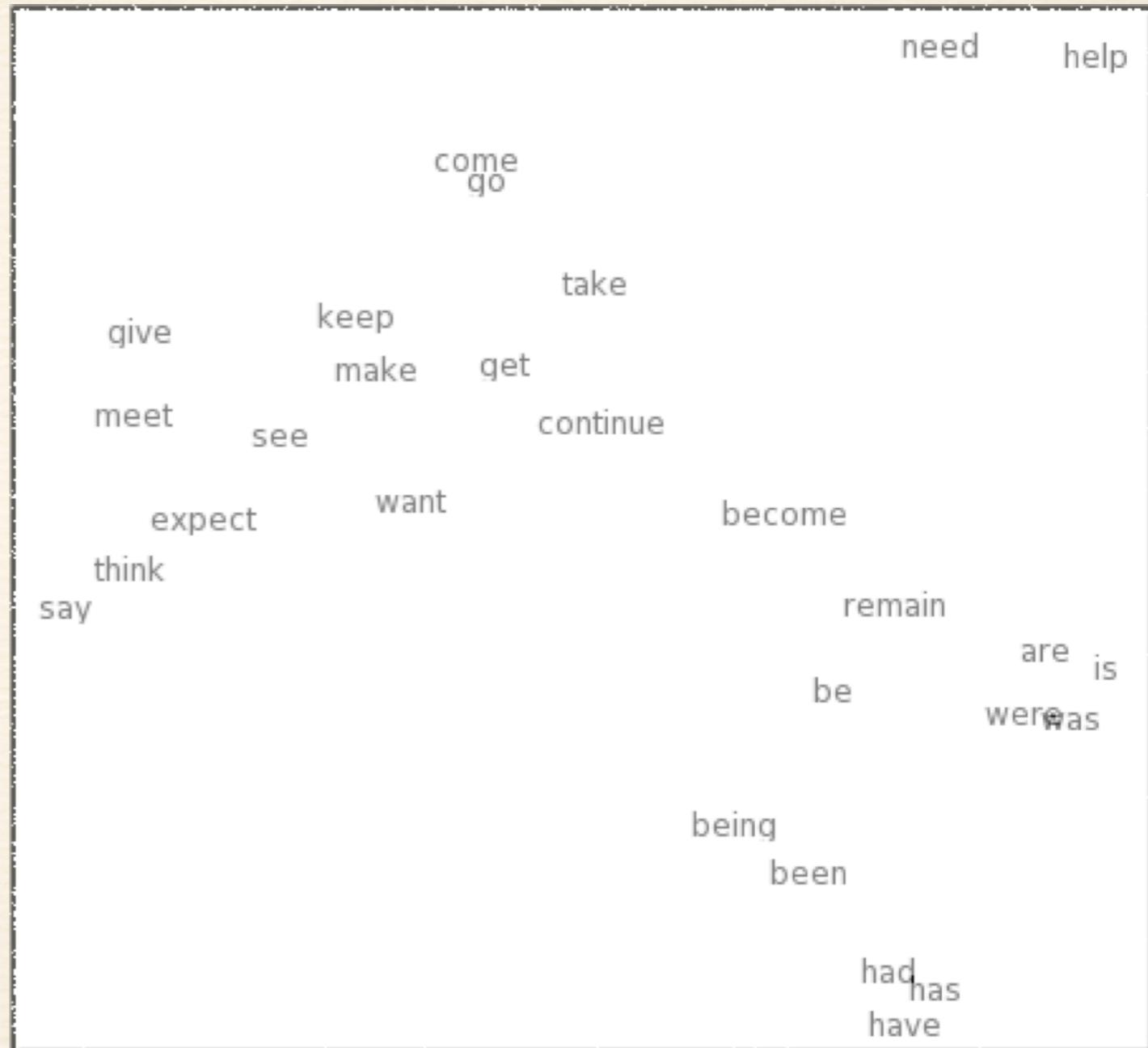
- ❖ Vector dimension: the number of words in vocabulary (e.g., 500,000+)
- ❖ **Problem:** For any two different words, their one-hot vectors are **orthogonal**. There is no natural notion of **similarity** for one-hot vectors
- ❖ How to encode similarity for words, so that we can, for example, match documents containing "Lafayette hotel", if a user searches for "Lafayette motel" in the web?

Word vectors (embeddings)

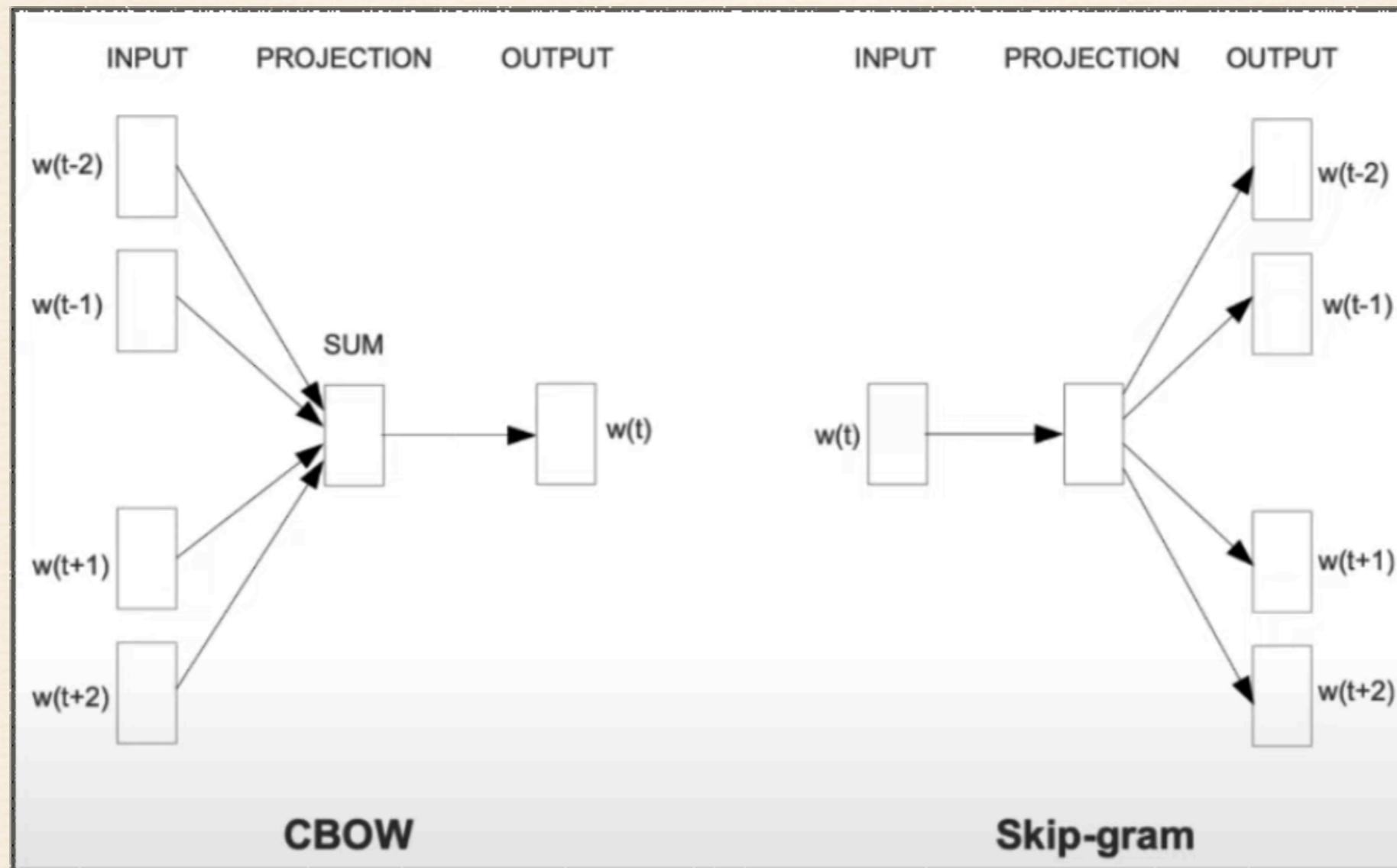
- ❖ Distributional semantics: A word's meaning is given by the **words that frequently appear close-by**
- ❖ When a word **w** appears in a text, its **context** is the set of words that appear nearby (within a fixed-size window)
- ❖ We use the many contexts of **w** to build up a representation of **w**
- ❖ More specifically, we will build a dense vector for each word, chosen/learned so that it is similar to vectors of words that appear in similar contexts, measuring similarity as the vector dot (scalar) product

Word vectors (embeddings)

- ❖ Word2Vec (Mikolov et al., 2013) by Google
- ❖ GloVe (Pennington et al., 2014), Stanford
- ❖ **Learn** numeric representation of the meanings of words



Word2Vec



Continuous Bag of Words model
Predict center word
from context words

Skip-gram model
Predict context words
from center word

Word2Vec encodes meaning into vectors,
but what if a word has different meaning
in different sentences?

*The **bank** of the river*

vs.

*Money in the **bank***

Attention

transformer.huggingface.co/doc/gpt2-large Incognito

Write With Transformer gpt2 ⓘ

Shuffle initial text Trigger autocomplete or tab Select suggestion ↑ ↓ and enter Cancel suggestion esc

Save & Publish

generate word by word

As aliens entered our planet and began to colonized Earth, a certain group of extraterrestrials began to manipulate our society through their influences of a certain number of the elite to keep and iron grip over the populace.

Learns what words to attend to, when generating each word

As aliens entered our planet and began to colonize earth a certain group of extraterrestrials ...

```
graph TD; A([As]) --> As[As aliens entered our planet]; B([aliens]) --> Aliens[aliens]; C([entered]) --> Entered[entered]; D([our]) --> Our[our]; E([planet]) --> Planet[planet]; F([and]) --> And[and]; G([began]) --> Began[began]; H([to]) --> To[to]; I([colonize]) --> Colonize[colonize earth]; J([earth]) --> Earth[earth]; K([a]) --> A[A certain]; L([certain]) --> Certain[certain]; M([group]) --> Group[group]; N([of]) --> Of[of]; O([extraterrestrials]) --> Extraterrestrials[extraterrestrials]; P([...]) --> Ellipsis[...]
```

Recurrent Neural Networks has a short reference window

As aliens entered our planet

and began to colonize earth a certain group of extraterrestrials ...



GRU's and LSTM's have a longer reference window than RNN's

As aliens entered our planet

and began to colonize earth a certain group of extraterrestrials ...

Transformer

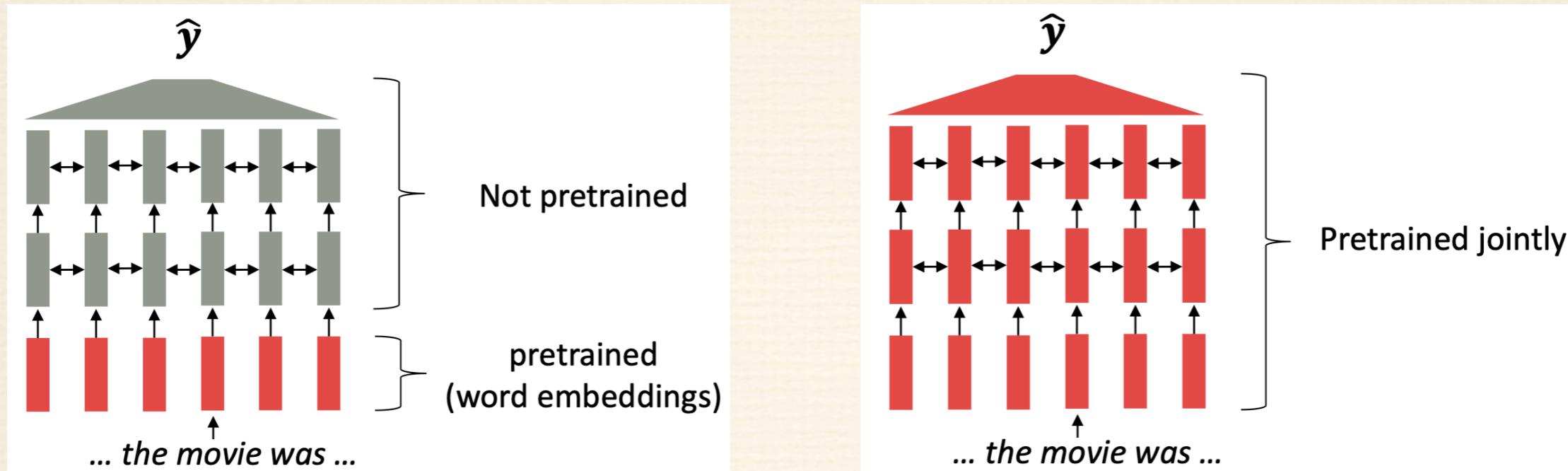
Attention Mechanism has an infinite reference window
in theory, given enough compute resources

As aliens entered our planet

and began to colonize earth a certain group of extraterrestrials ...

Pretraining

- ❖ Pretained word embeddings (2017) -> pretrained whole model



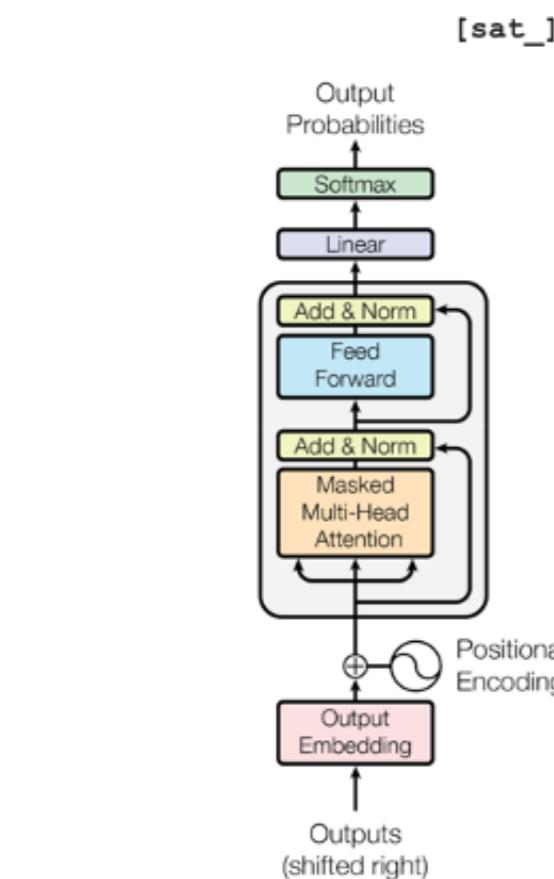
- ❖ All (or almost all) parameters in NLP networks are initialized via pretraining
- ❖ Pretraining methods hide parts of the input from the model, and train the model to reconstruct those parts
- ❖ Exceptionally effective at building strong representations of language, parameter initializations for strong NLP models, and probability distributions over language that we can sample from

Pretraining and finetuning

- ❖ Why should they help, from a “training neural networks” perspective?
 - ❖ Consider, provides parameters $\hat{\theta}$ by approximating $\min_{\theta} L_{pretrain}(\theta)$.
(The pretraining loss)
 - ❖ Then, finetuning approximates $\min_{\theta} L_{finetune}(\theta)$, starting at $\hat{\theta}$. (The finetuning loss)
- ❖ The pretraining may matter because stochastic gradient descent sticks (relatively) close to $\hat{\theta}$ during finetuning.
 - ❖ So, maybe the finetuning local minima near $\hat{\theta}$ tend to generalize well
 - ❖ And/or, maybe the gradients of finetuning loss near $\hat{\theta}$ propagate nicely

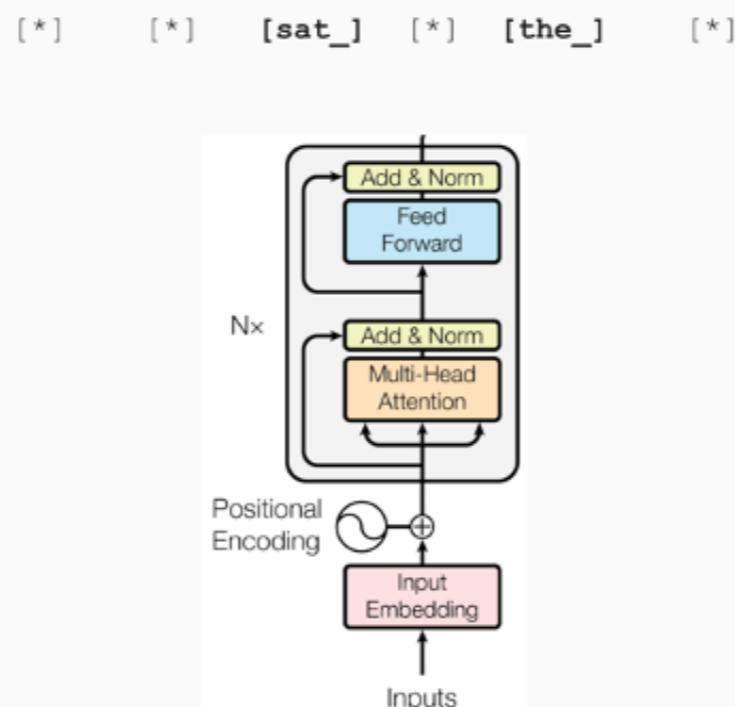
Recall: Notable LLMs

Decoder-only GPT



[START] [The_] [cat_]

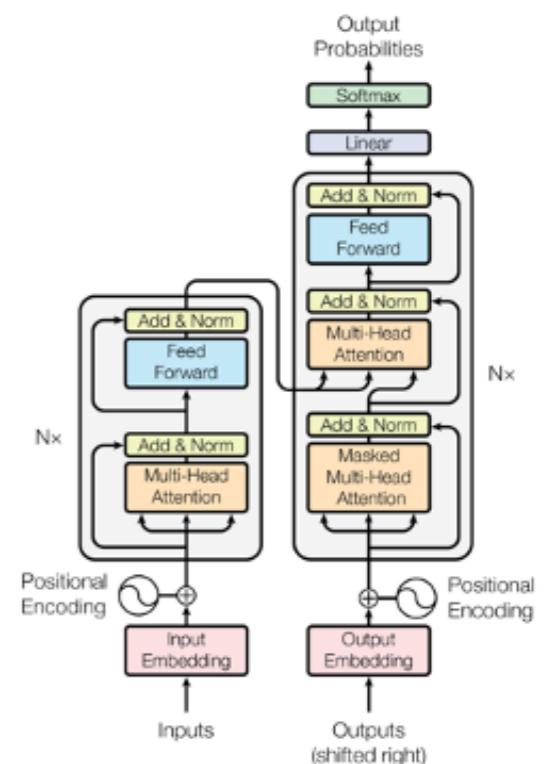
Encoder-only BERT



[The_] [cat_] [MASK] [on_] [MASK] [mat_]

Enc-Dec T5

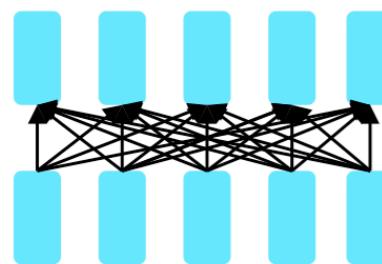
Das ist gut.
A storm in Attala caused 6 victims.
This is not toxic.



Translate EN-DE: This is good.
Summarize: state authorities dispatched...
Is this toxic: You look beautiful today!

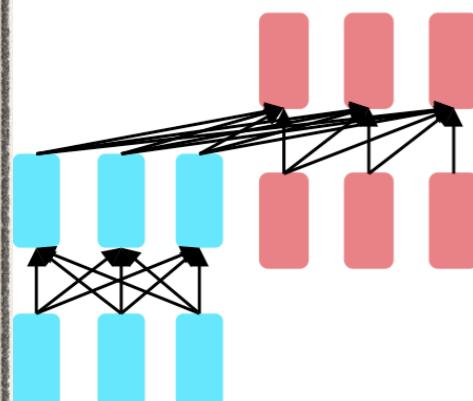
Pretraining for 3 types of architectures

- ❖ The neural architecture influences the type of pretraining, and natural use cases



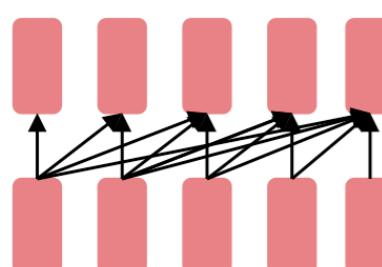
Encoders

- Gets bidirectional context – can condition on future!
- How do we train them to build strong representations?



Encoder-Decoders

- Good parts of decoders and encoders?
- What's the best way to pretrain them?

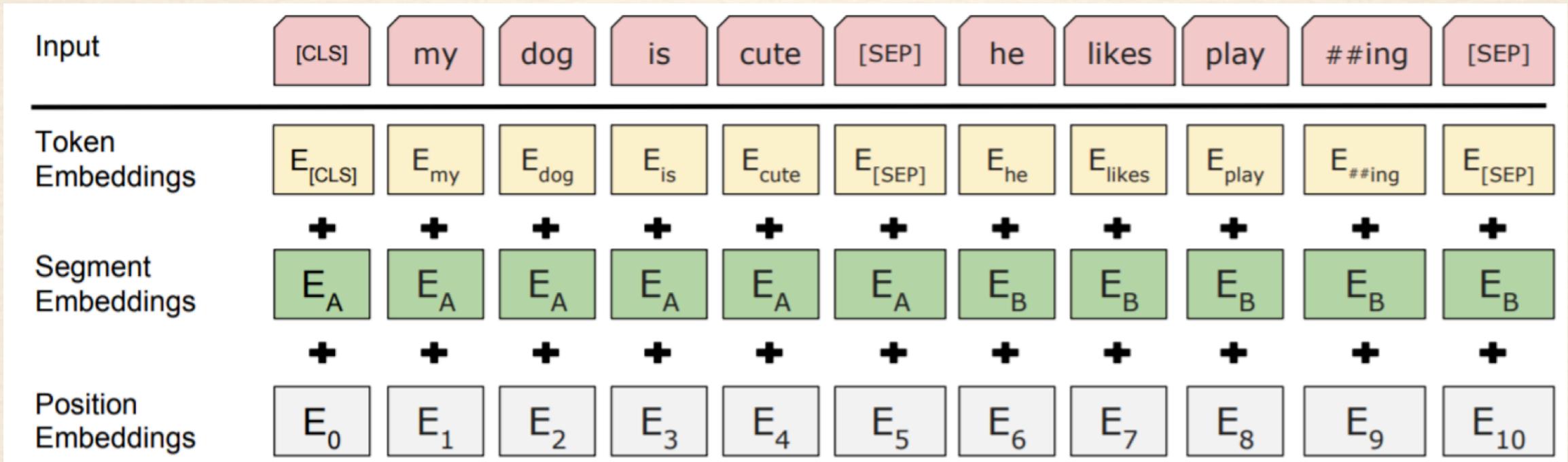


Decoders

- Language models! What we've seen so far.
- Nice to generate from; can't condition on future words

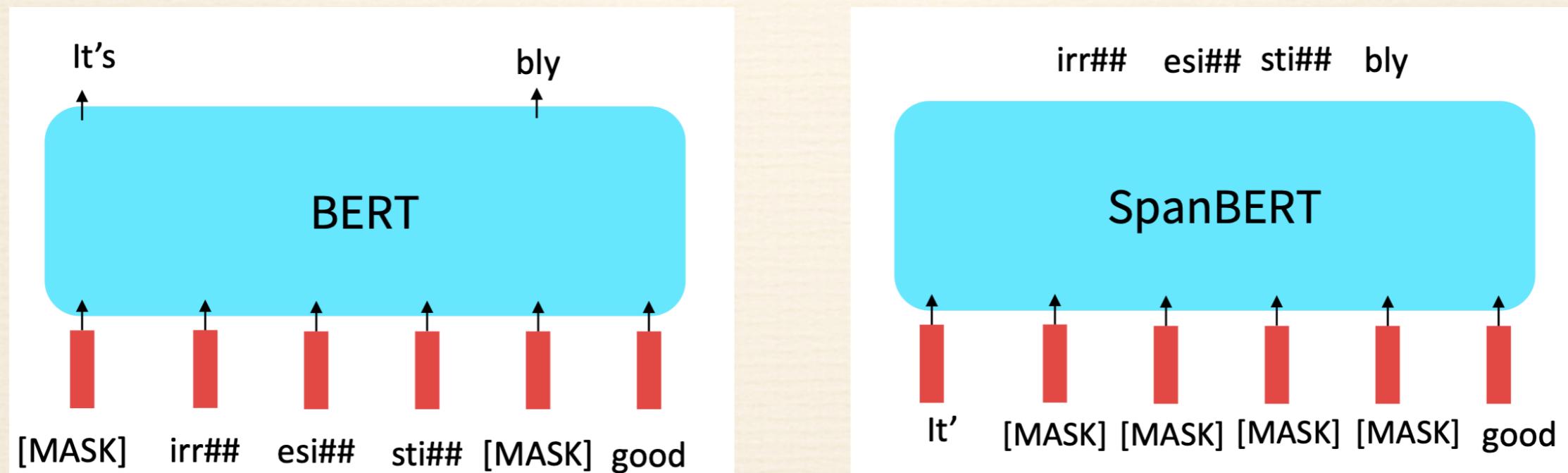
Pretraining encoders

- ❖ What pretraining objectives to use?
- ❖ Idea: replace some fraction of words in the input with a special [MASK] token; predict these words
- ❖ BERT: Bidirectional Encoder Representations from Transformers, Devlin et al., 2018
 - ❖ Two objectives: Masked language modeling and Next sentence prediction
 - ❖ Trained on: BooksCorpus (800 million words), English Wikipedia (2,500 million words)
 - ❖ BERT-base: 12 layers, 768-dim hidden states, 12 attention heads, 110 million params
 - ❖ BERT-large: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million params

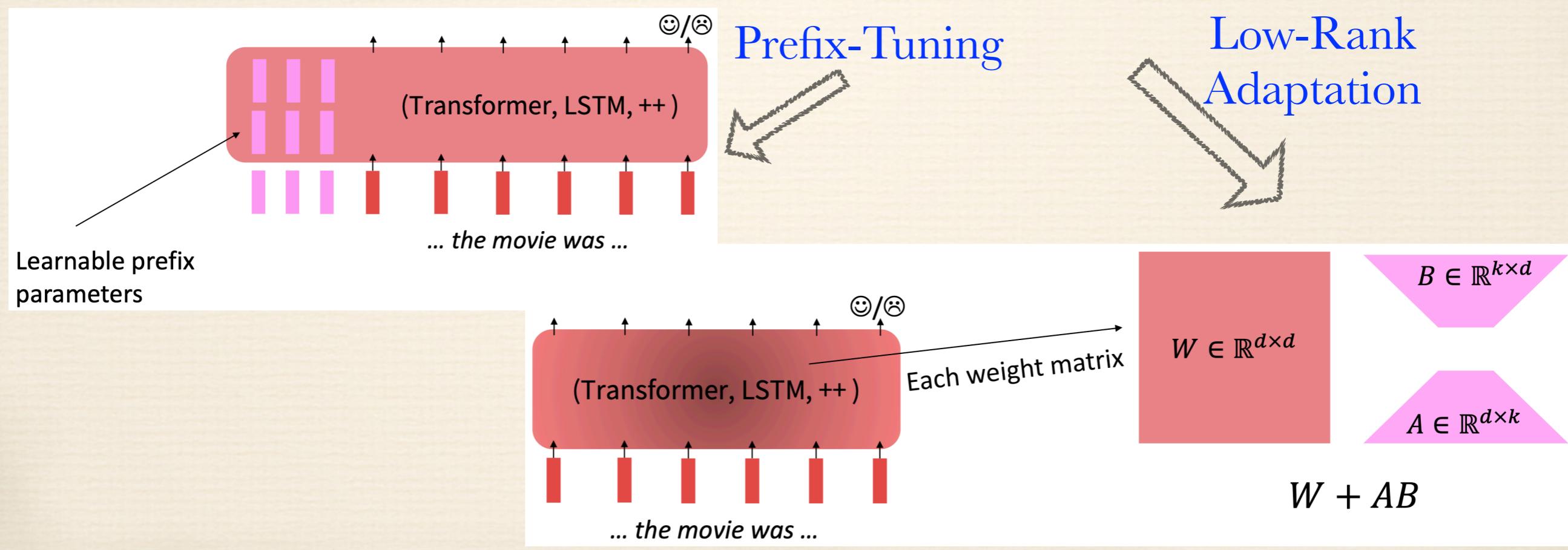
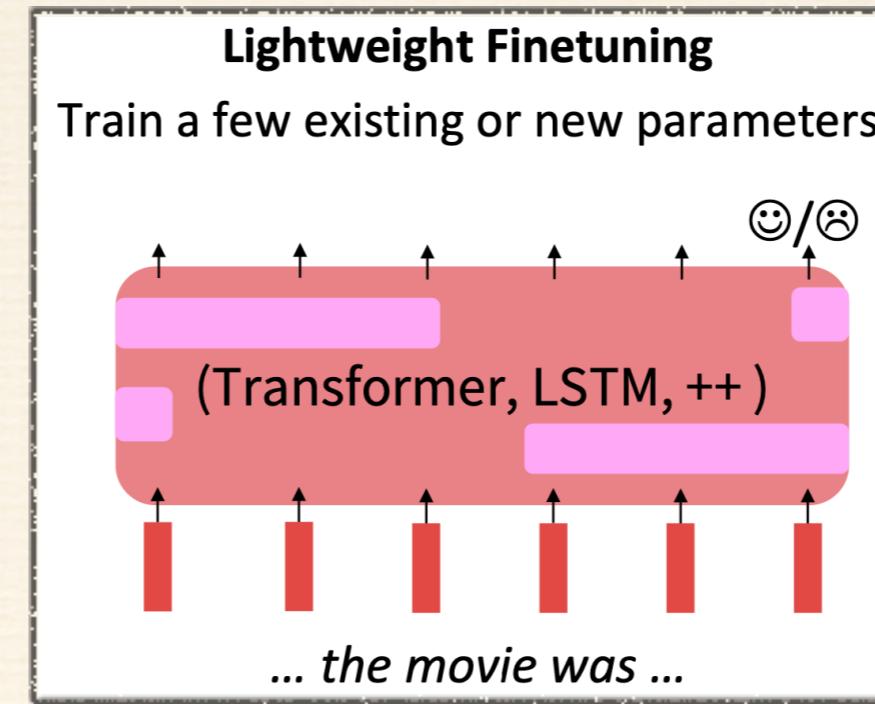
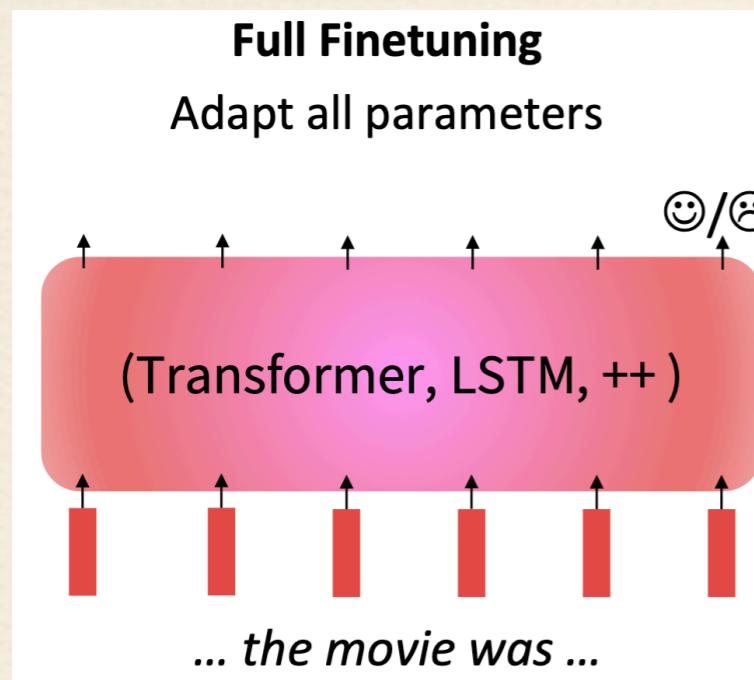


Extensions of BERT

- ❖ Some generally accepted improvements to the BERT pretraining formula:
 - ❖ RoBERTa (Liu et al., 2019): mainly just train BERT for **longer** and **remove** next sentence prediction
 - ❖ SpanBERT (Joshi et al., 2020): masking **contiguous spans** of words makes a harder, more useful pretraining task

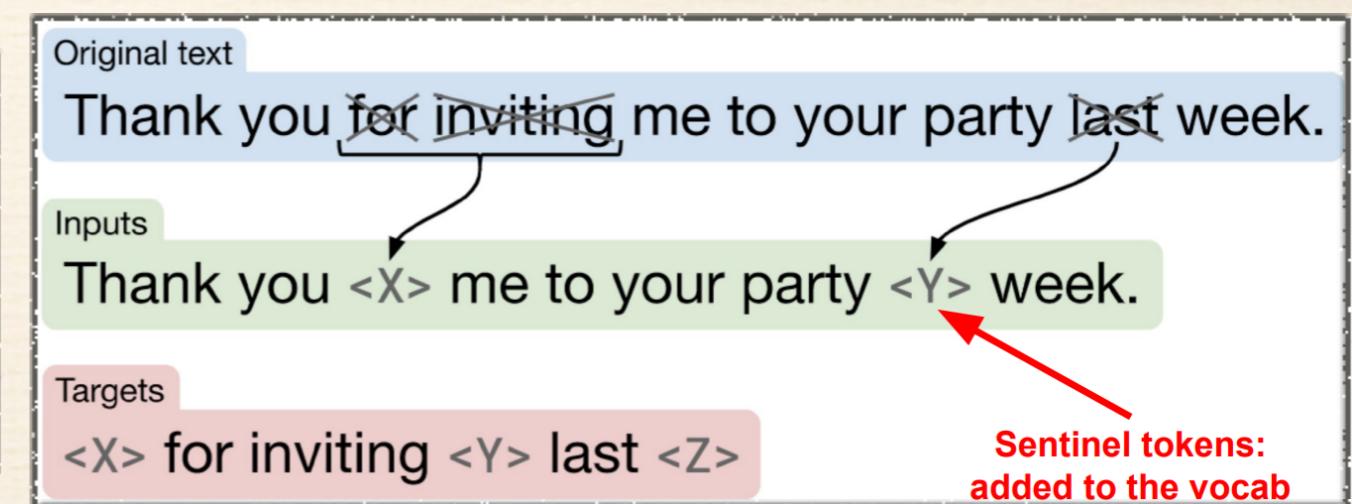
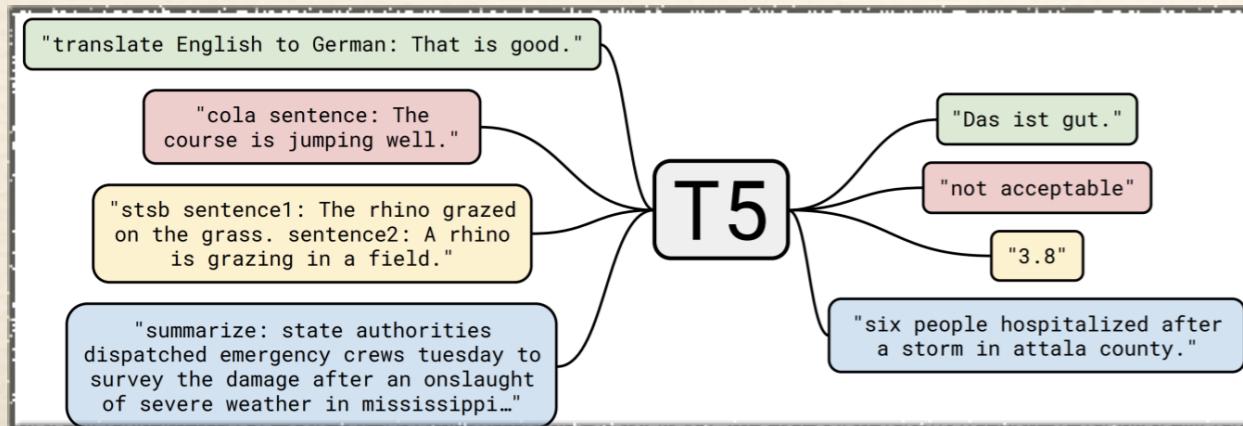


Full vs. Parameter-Efficient Finetuning



Pretraining encoder-decoders

- ❖ What pretraining objectives to use?
- ❖ Idea: replace different-length spans from the input with unique placeholders; decode out the spans that were removed
- ❖ Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, Raffel et al., 2018
 - ❖ T5: Text-to-Text Transfer Transformer, 11 Billion parameters
 - ❖ Pertaining on Colossal Clean Crawled Corpus (C4): start with Common Crawl (over 50TB of compressed data, 10B+ web pages), filtered down to ~800GB, or ~160B tokens
 - ❖ trained with a novel text infilling objective: randomly mask a portion of contiguous tokens and train the model to predict the masked text spans



Text-to-text framework

Pretraining objective

Pretraining decoders

- ❖ It's natural to pretrain decoders as language models and then use them as generators
- ❖ We can finetune them by training a classifier on the last word's hidden state
- ❖ Improving Language Understanding by Generative Pre-Training, Radford et al., 2018
 - ❖ Transformer decoder with 12 layers, 117M parameters
 - ❖ 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers
 - ❖ Trained on BooksCorpus: over 7000 unique books
 - ❖ Contains long spans of contiguous text, for learning long-distance dependencies
 - ❖ The acronym “GPT” never showed up in the original paper; it could stand for “Generative PreTraining” or “Generative Pretrained Transformer”

Generative Pretrained Transformer (GPT)

- ❖ How do we format inputs to our decoder for finetuning tasks?
- ❖ Natural Language Inference: Label pairs of sentences as entailing/contradictory/neutral

Premise: *The man is in the doorway*
Hypothesis: *The person is near the door*

} entailment

- ❖ Here's roughly how the input was formatted, as a sequence of tokens for the decoder.

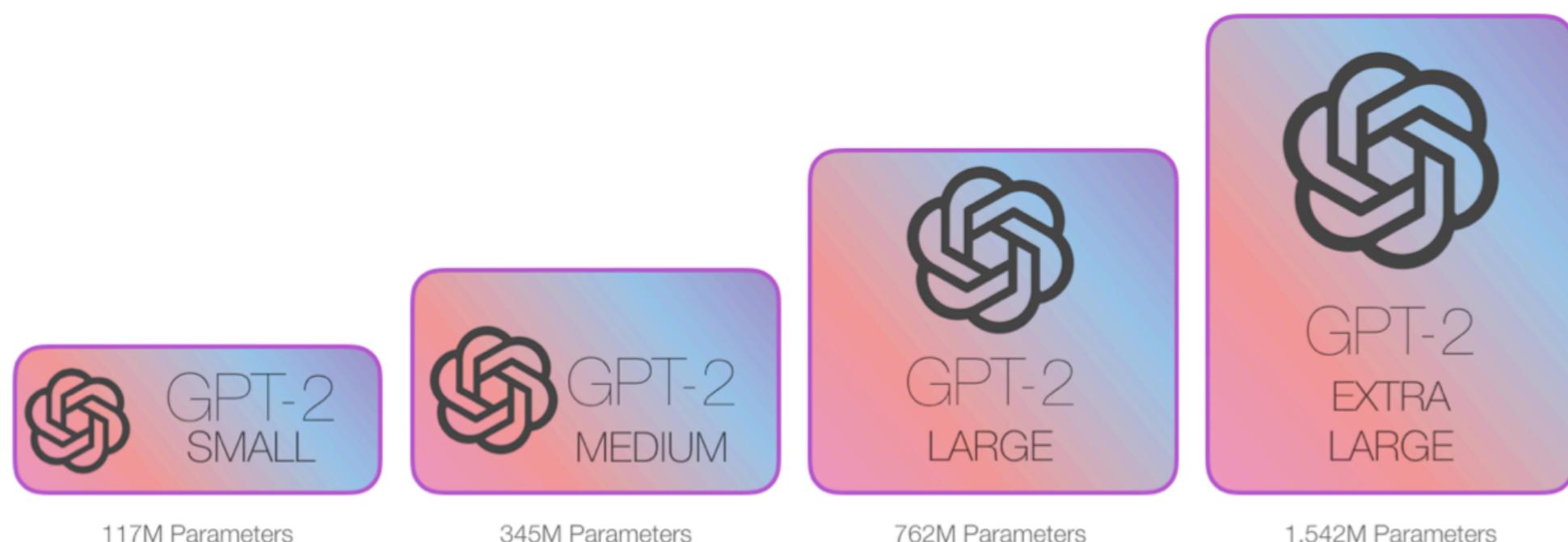
[START] *The man is in the doorway* [DELIM] *The person is near the door* [EXTRACT]

- ❖ The linear classifier is applied to the representation of the [EXTRACT] token.

GPT-2

- ❖ GPT-2, 2019, a larger version (1.5B) of GPT trained on more data
- ❖ Start to achieve strong zero-shot performance

Context size = 1024



.. trained on 40Gb of Internet text ..

GPT-3

- ❖ GPT-3 (Brown et al. 2020) further scaled the GPT-2 model to 175 Billion (100 times larger compared to the largest GPT-2 model), trained on 300B tokens of text.
- ❖ Before GPT-3, fine-tuning is the default way of doing learning in models like BERT/T5/GPT-2:
 - ❖ 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.

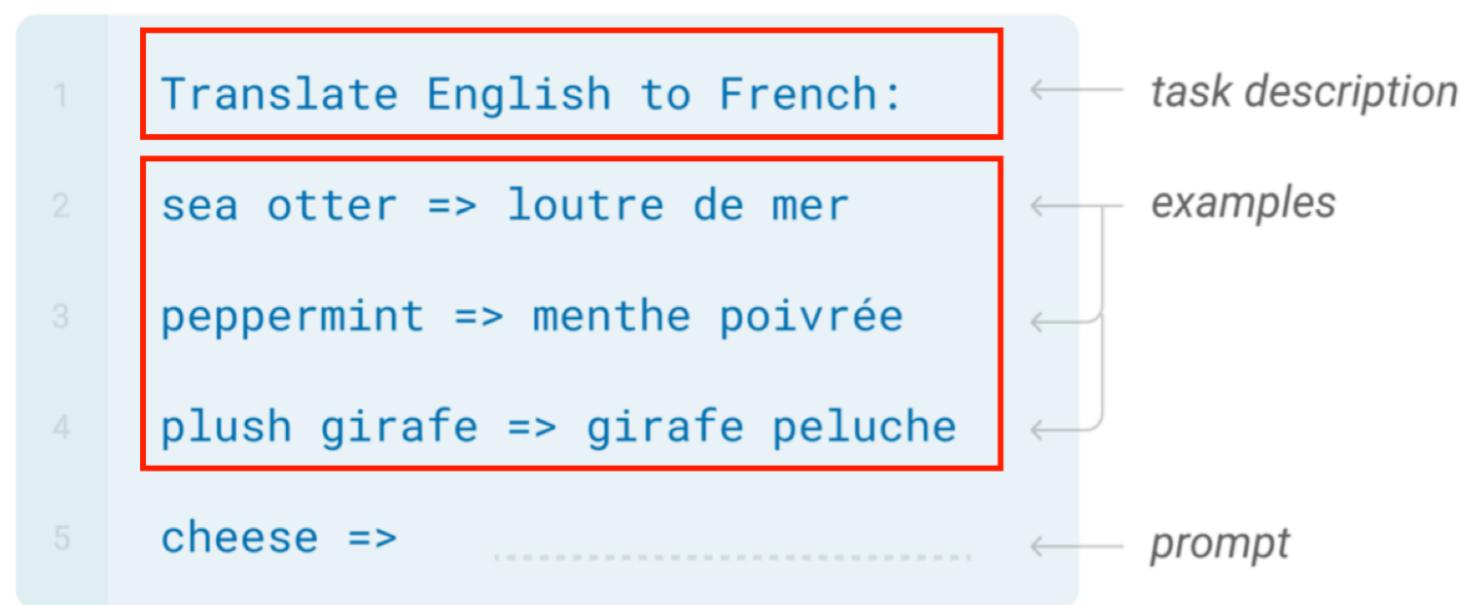


GPT-3 paradigm shift

- ❖ GPT-3 proposes an alternative: **In-context learning**
- ❖ Only need to feed a small number of examples (e.g., 32)
- ❖ Just a forward pass, **no gradient update** at all

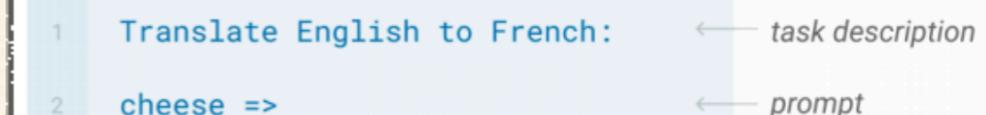
Few-shot

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



Zero-shot

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

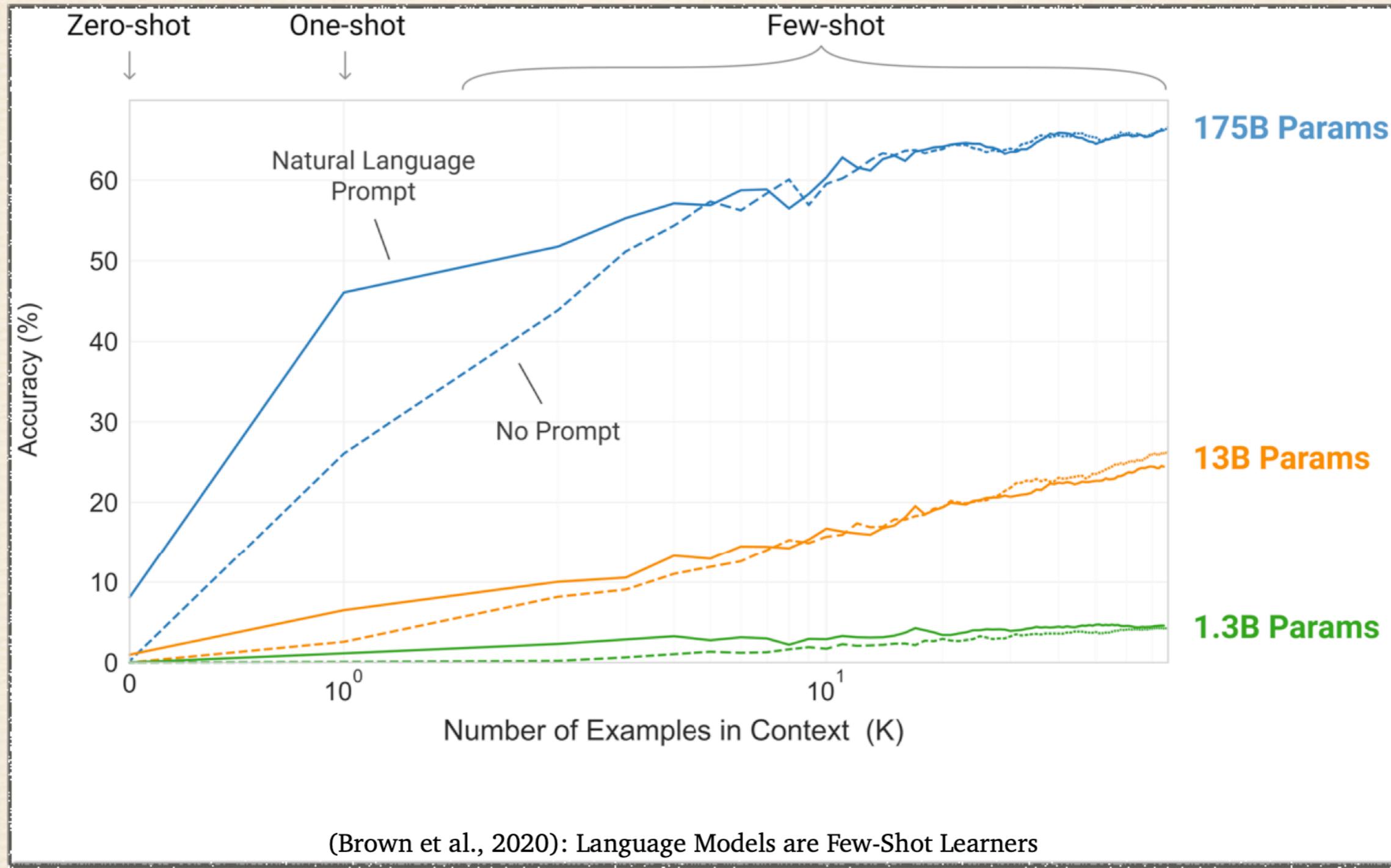


One-shot

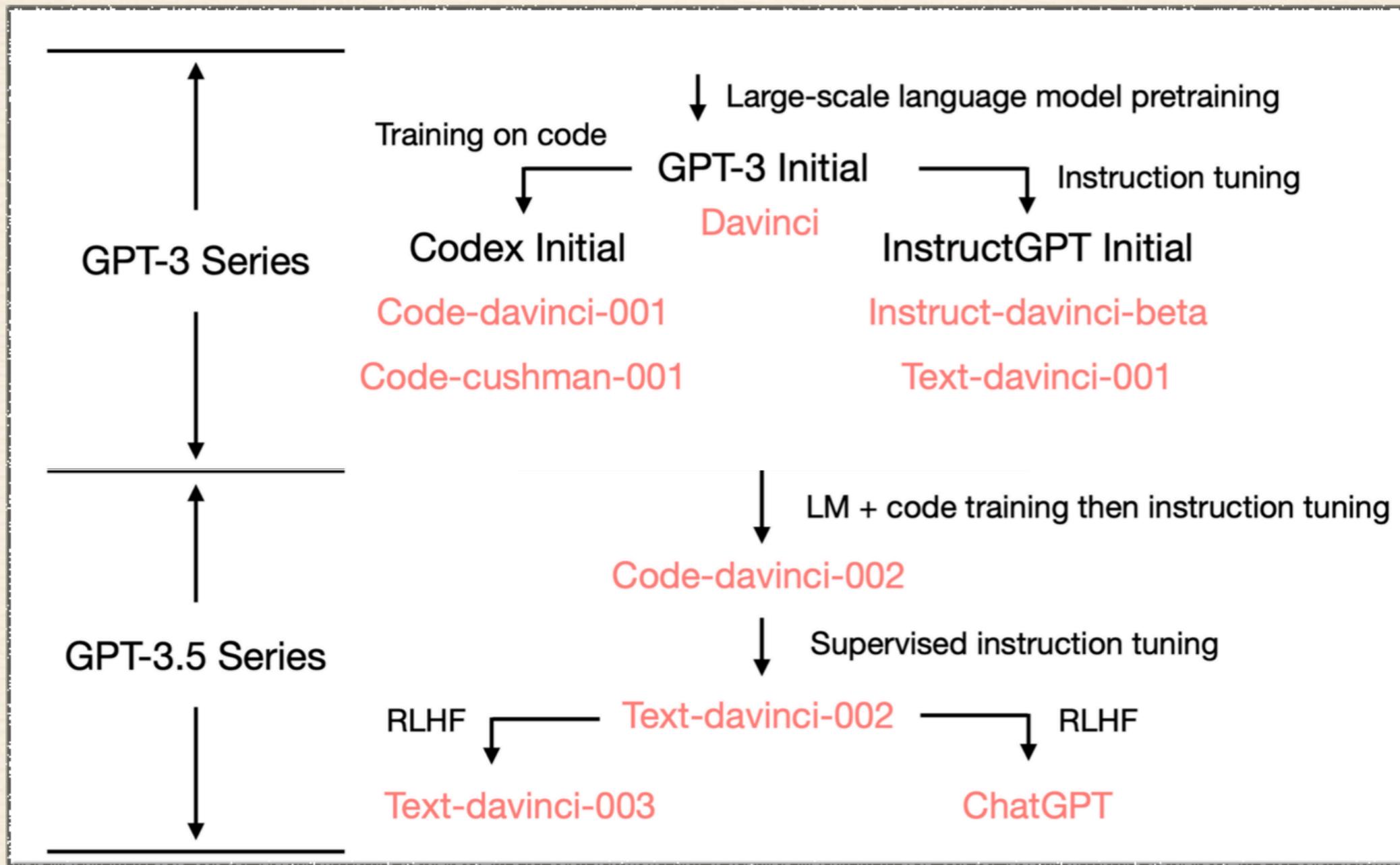
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



GPT-3's in-context learning



The GPT Lineage



New after GPT-3:
Training on code, supervised instruction tuning, and RLHF
(reinforcement learning from human feedback)

ChatGPT

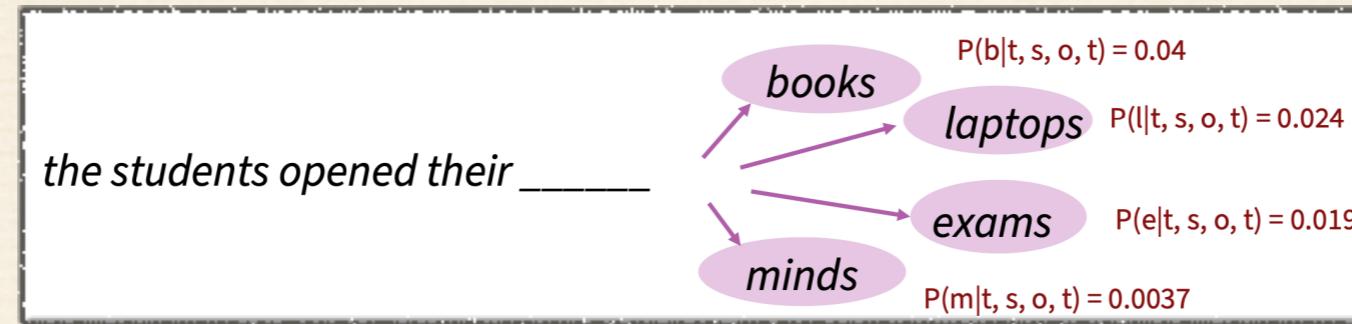
How Long it Took Top Apps to hit 100M Monthly Users



GPT-3.5 (Nov. 2022) -> GPT-4 (latest)

LMs to assistants

How do we get from language models

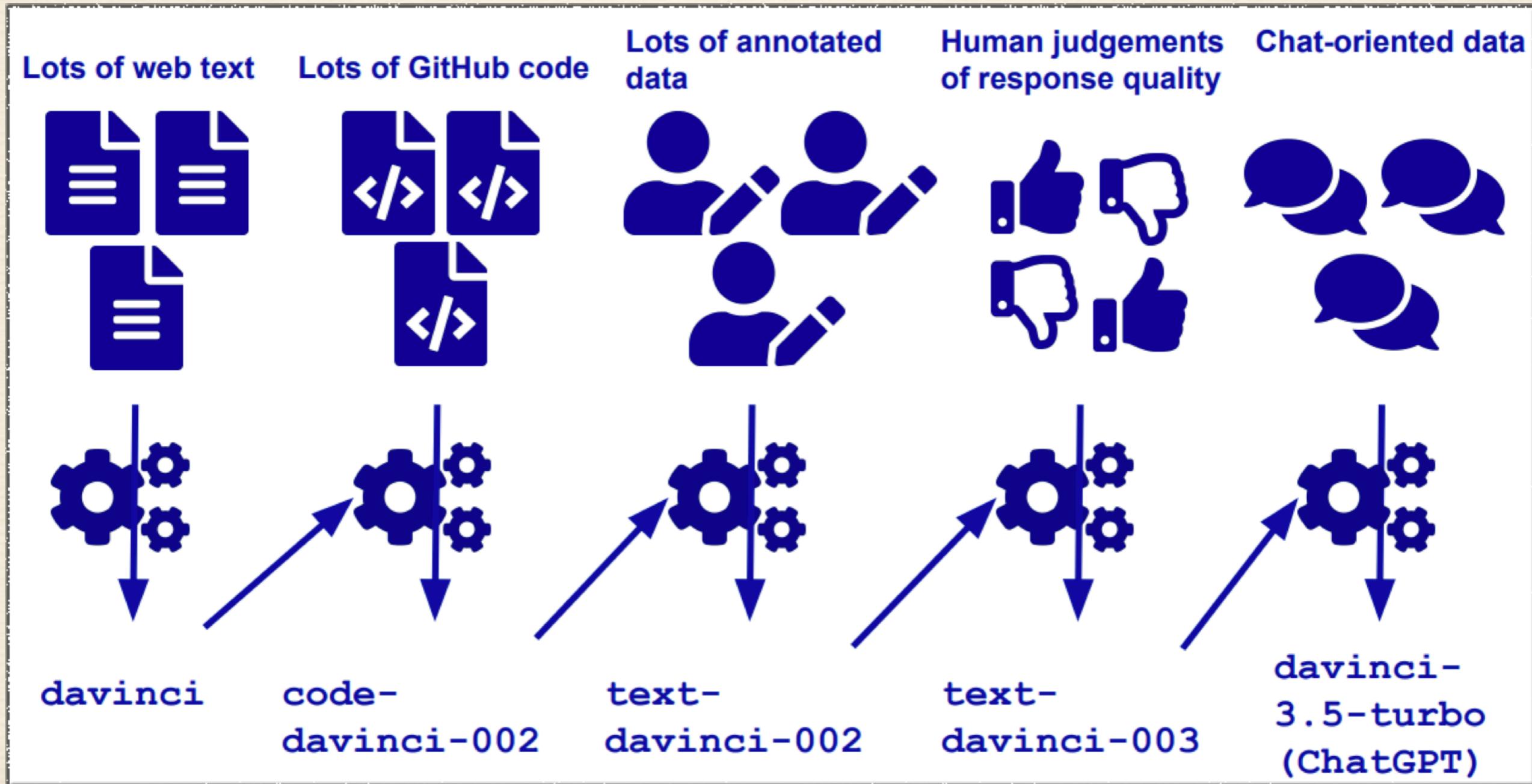


to this?

ChatGPT

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

ChatGPT



InstructGPT

Supervised instruction tuning + RLHF

Step 1

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

A prompt is sampled from our prompt dataset.

Explain reinforcement learning to a 6 year old.

A labeler demonstrates the desired output behavior.

We give treats and punishments to teach...

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

SFT

We give treats and punishments to teach...

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)

<https://huggingface.co/google/flan-t5-xxl>

[Scaling Instruction-Finetuned Language Models, 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). 

<https://huggingface.co/google/flan-t5-xxl>

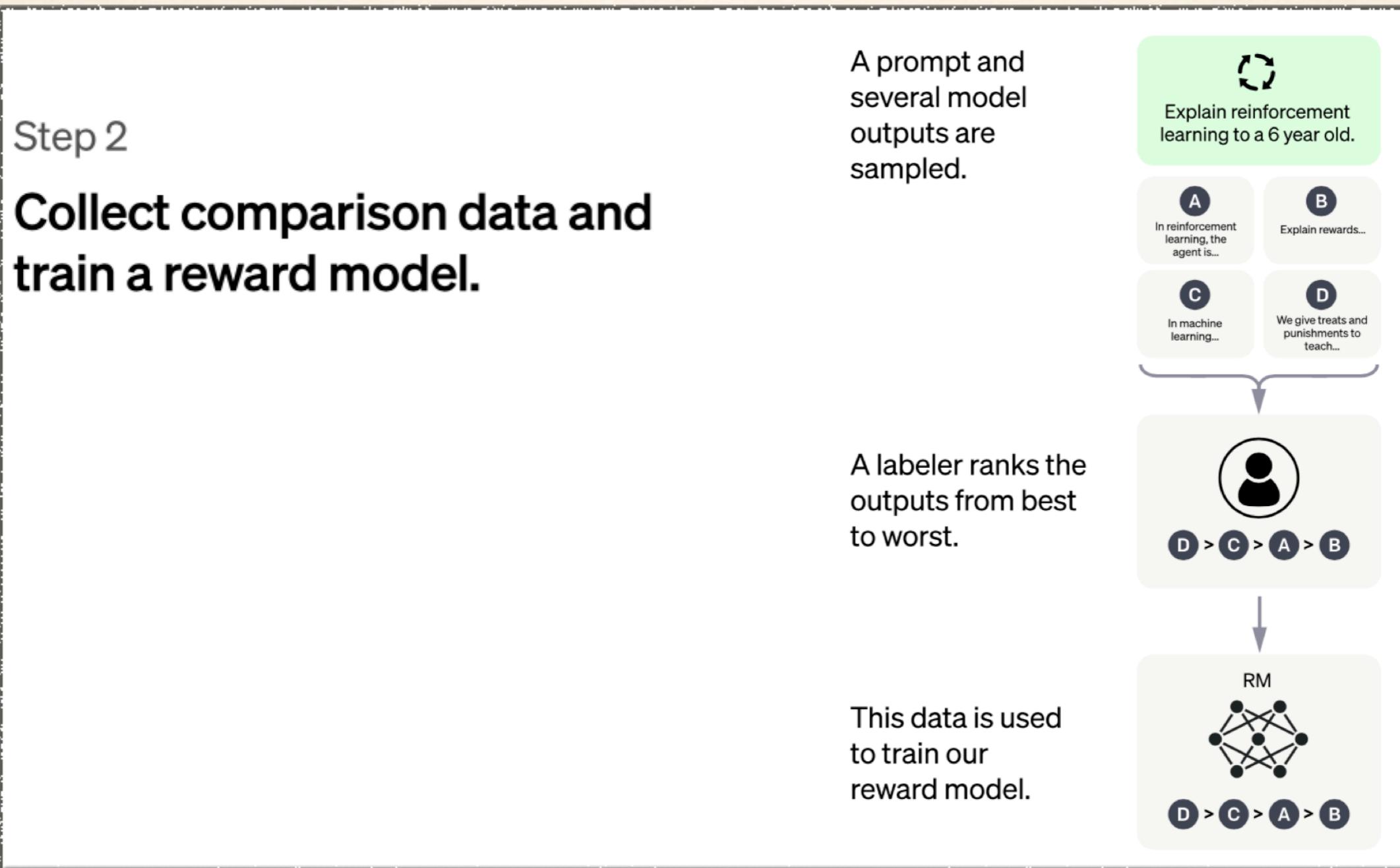
[Scaling Instruction-Finetuned Language Models, Chung et al. 2022]

Limitations

- ❖ It's expensive to collect ground-truth data for tasks.
- ❖ Tasks like open-ended creative generation have no right answer.
 - ❖ E.g., write me a story about a dog and her pet grasshopper.
- ❖ Language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
- ❖ Even with instruction finetuning, there a mismatch between the LM objective and the objective of “satisfy human preferences”.
- ❖ Can we explicitly attempt to satisfy human preferences?
 - ❖ Reinforcement learning from human feedback!

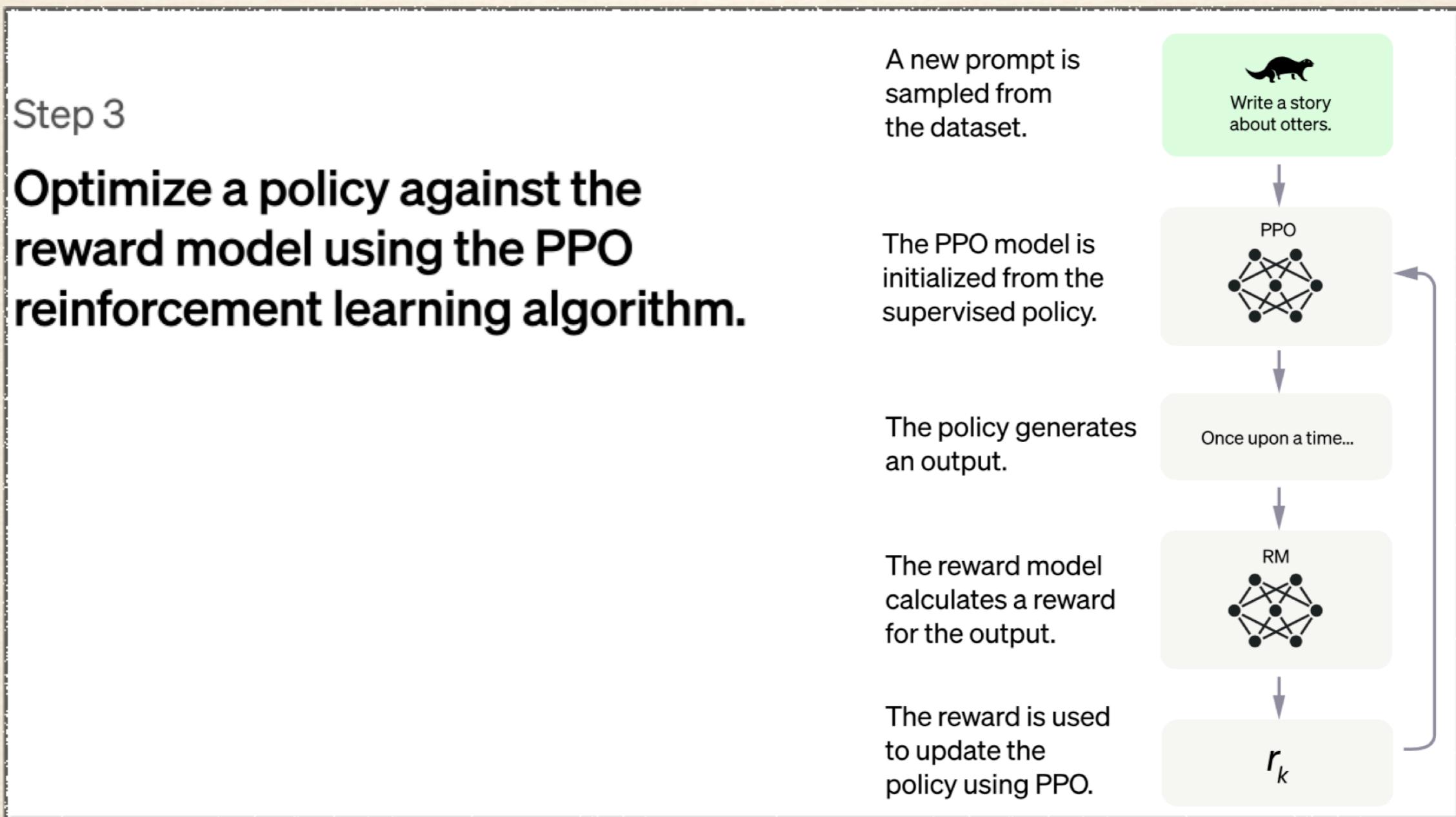
InstructGPT

Supervised instruction tuning + RLHF



InstructGPT

Supervised instruction tuning + RLHF



ChatGPT: InstructGPT + dialogue data

Introducing ChatGPT

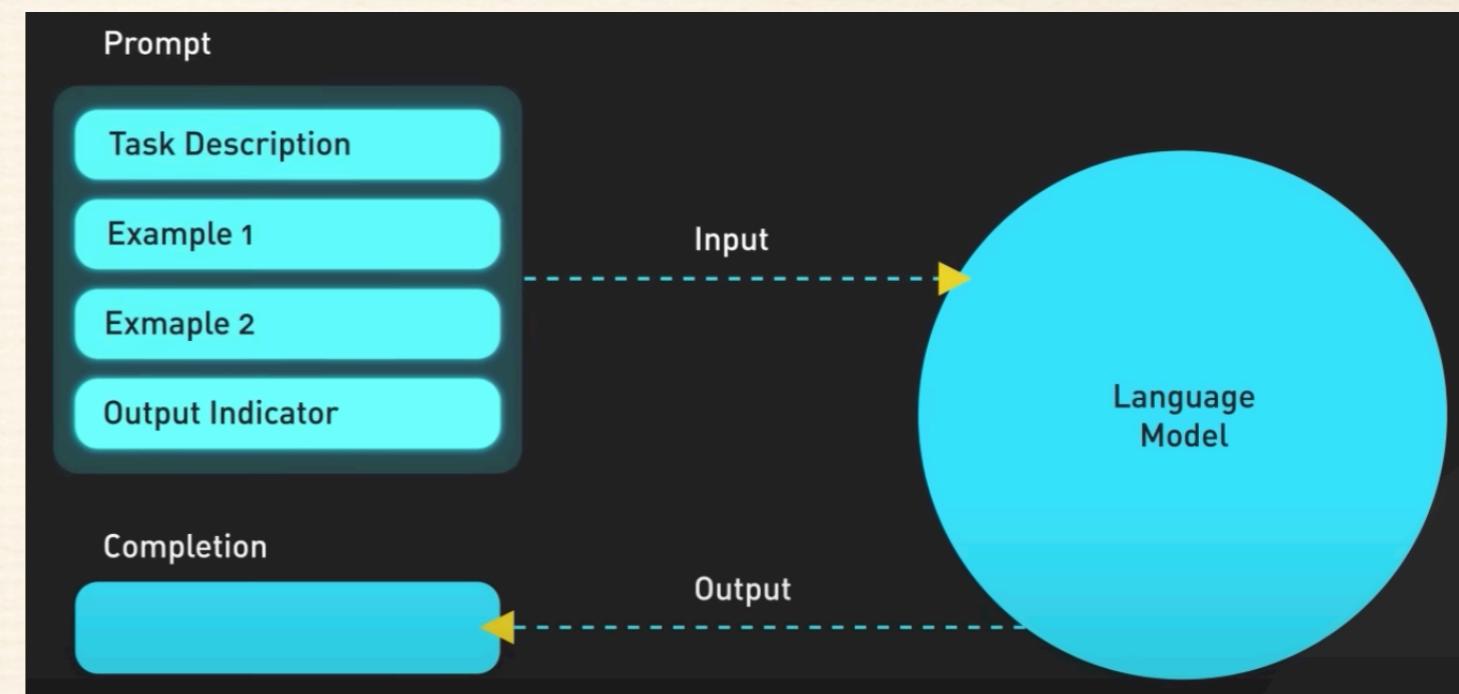
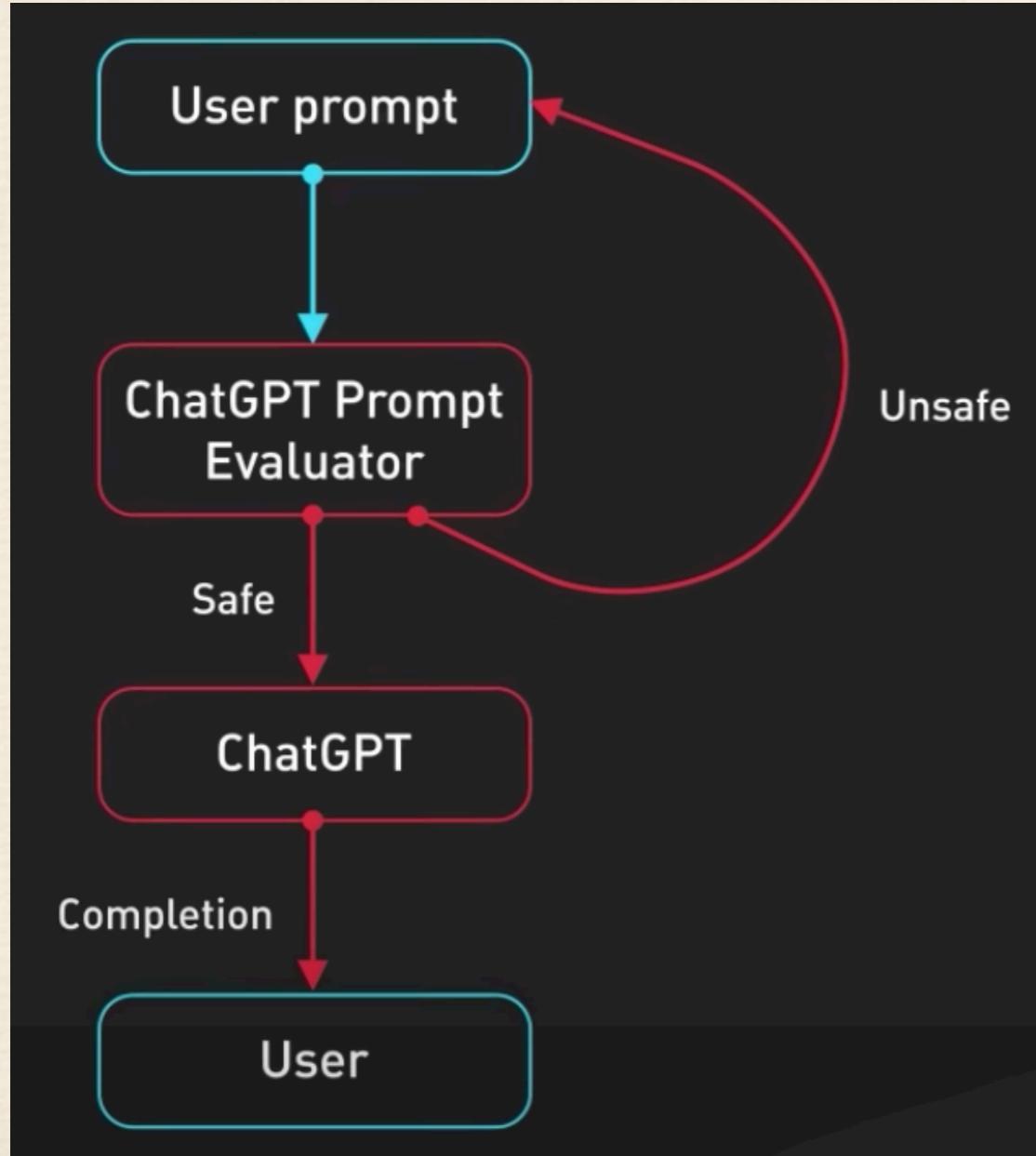
We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

"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."

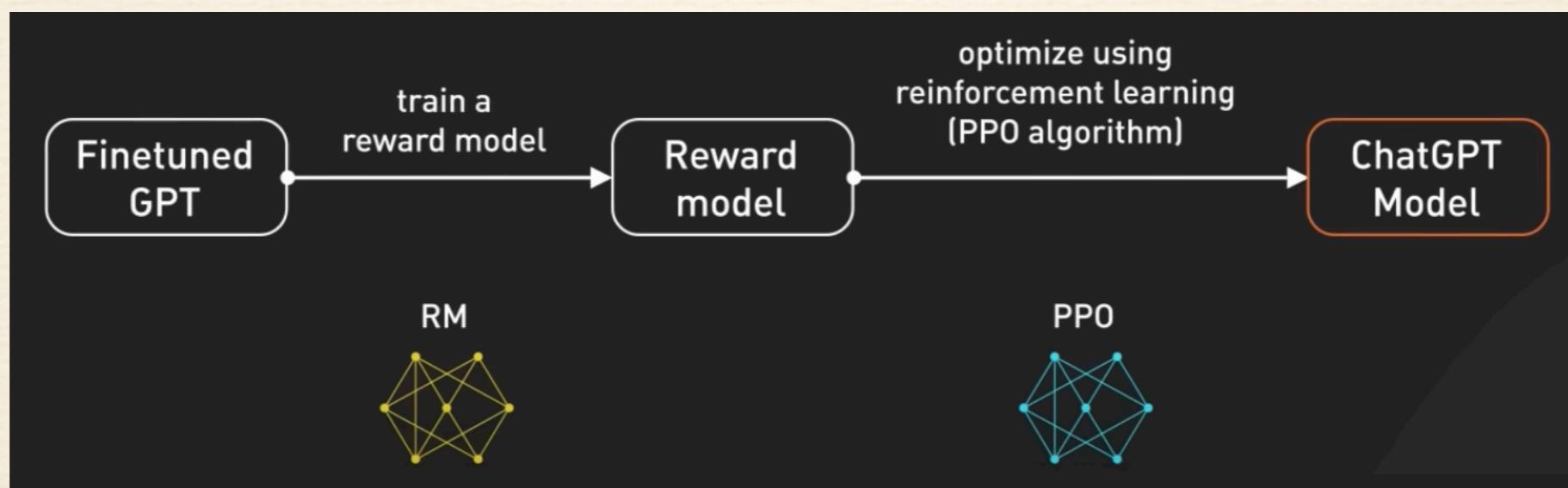
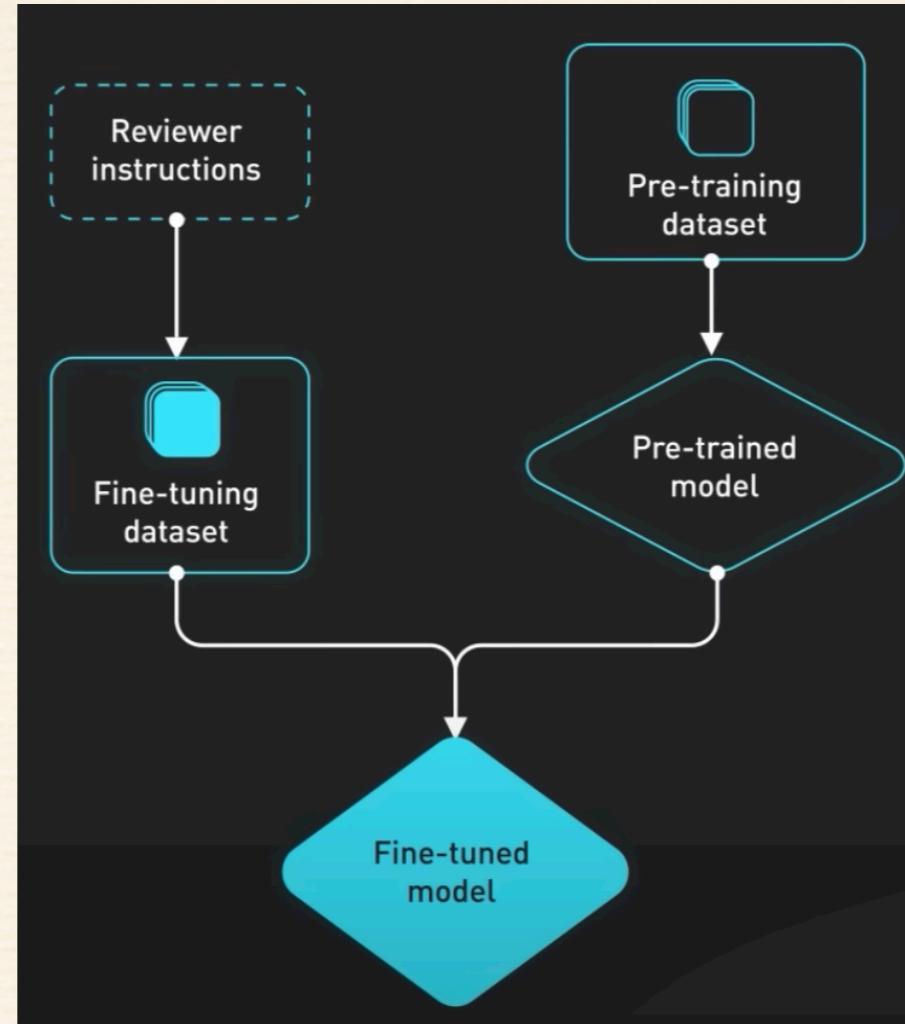
Human feedback data is the key!

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

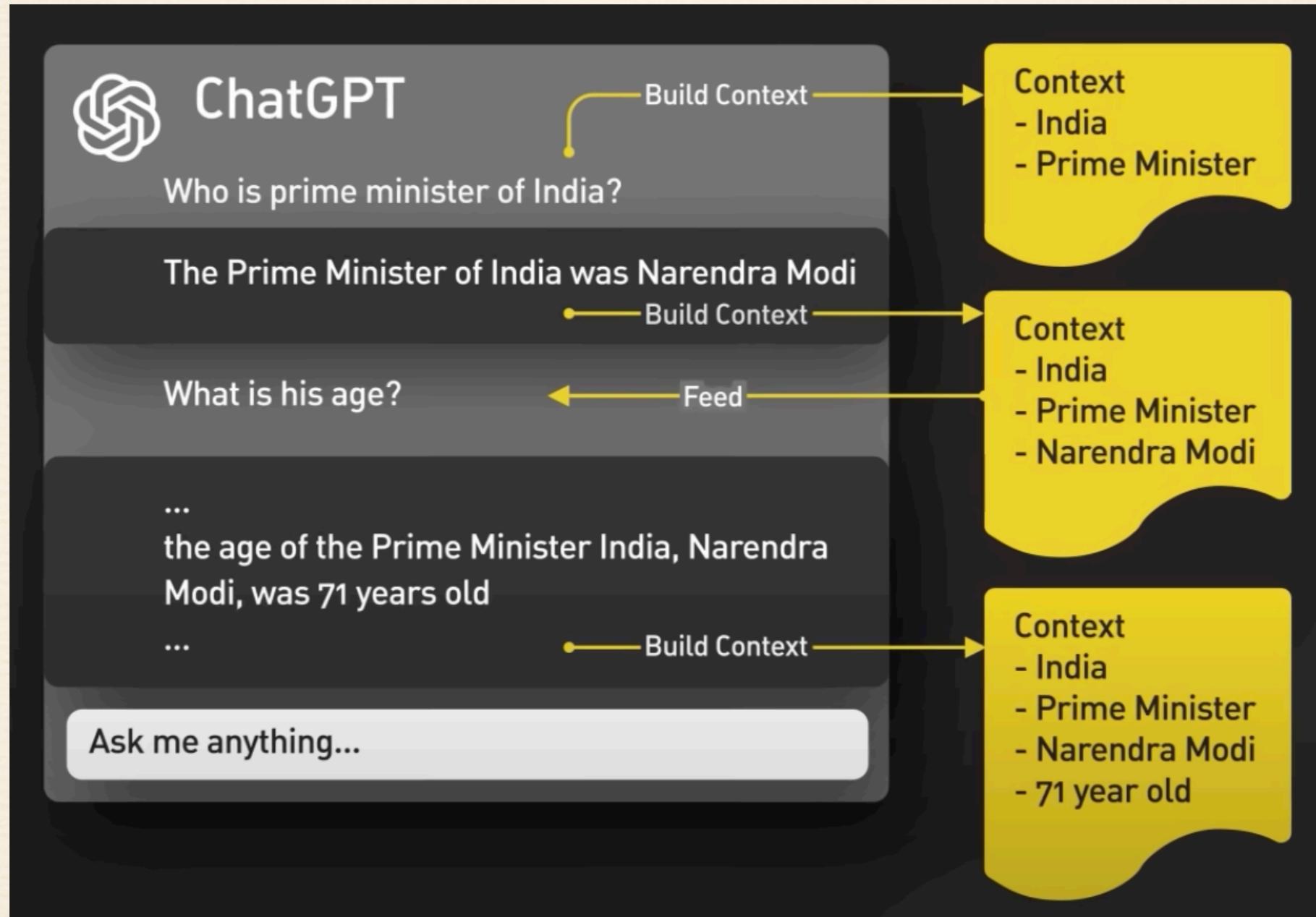
Summary



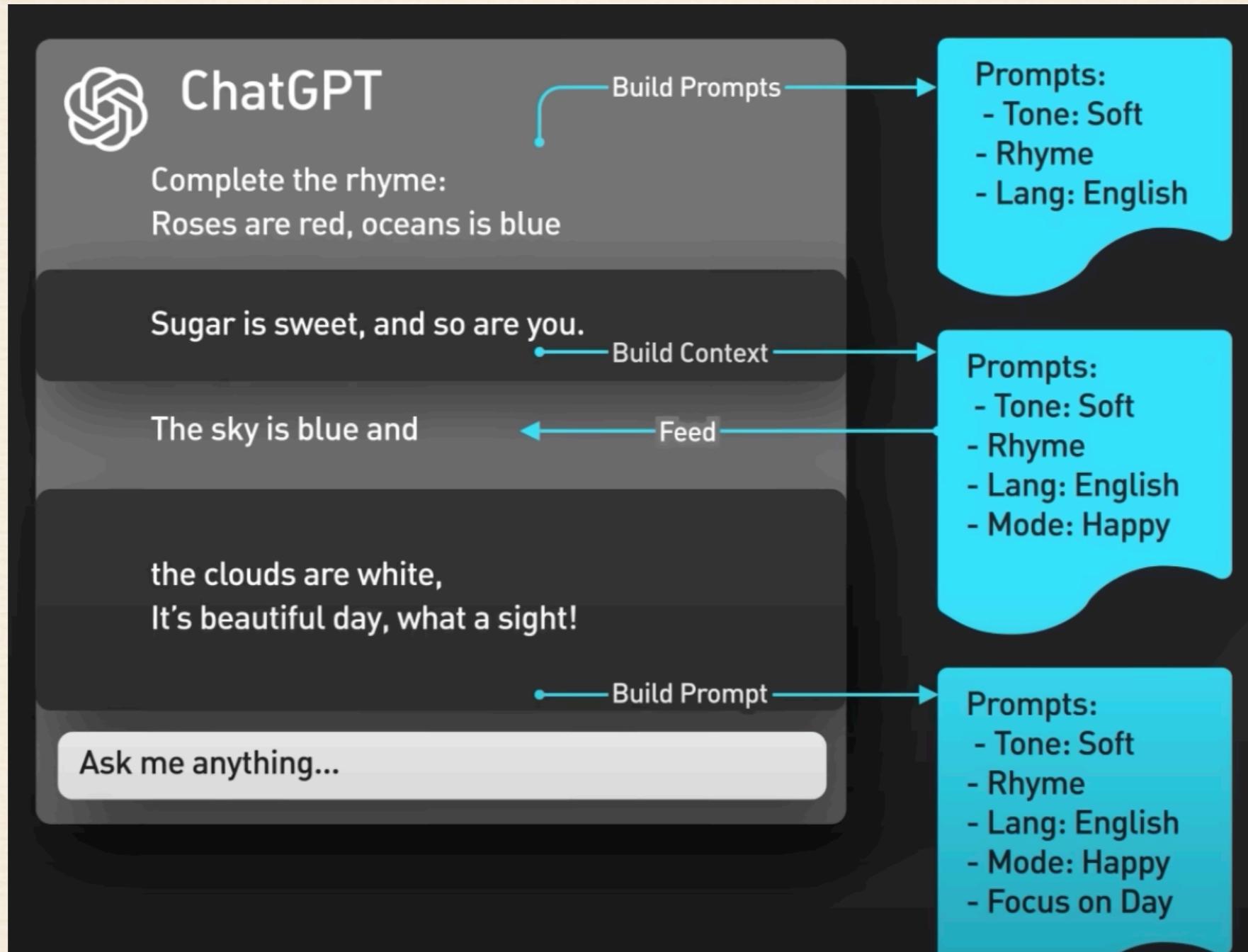
Summary



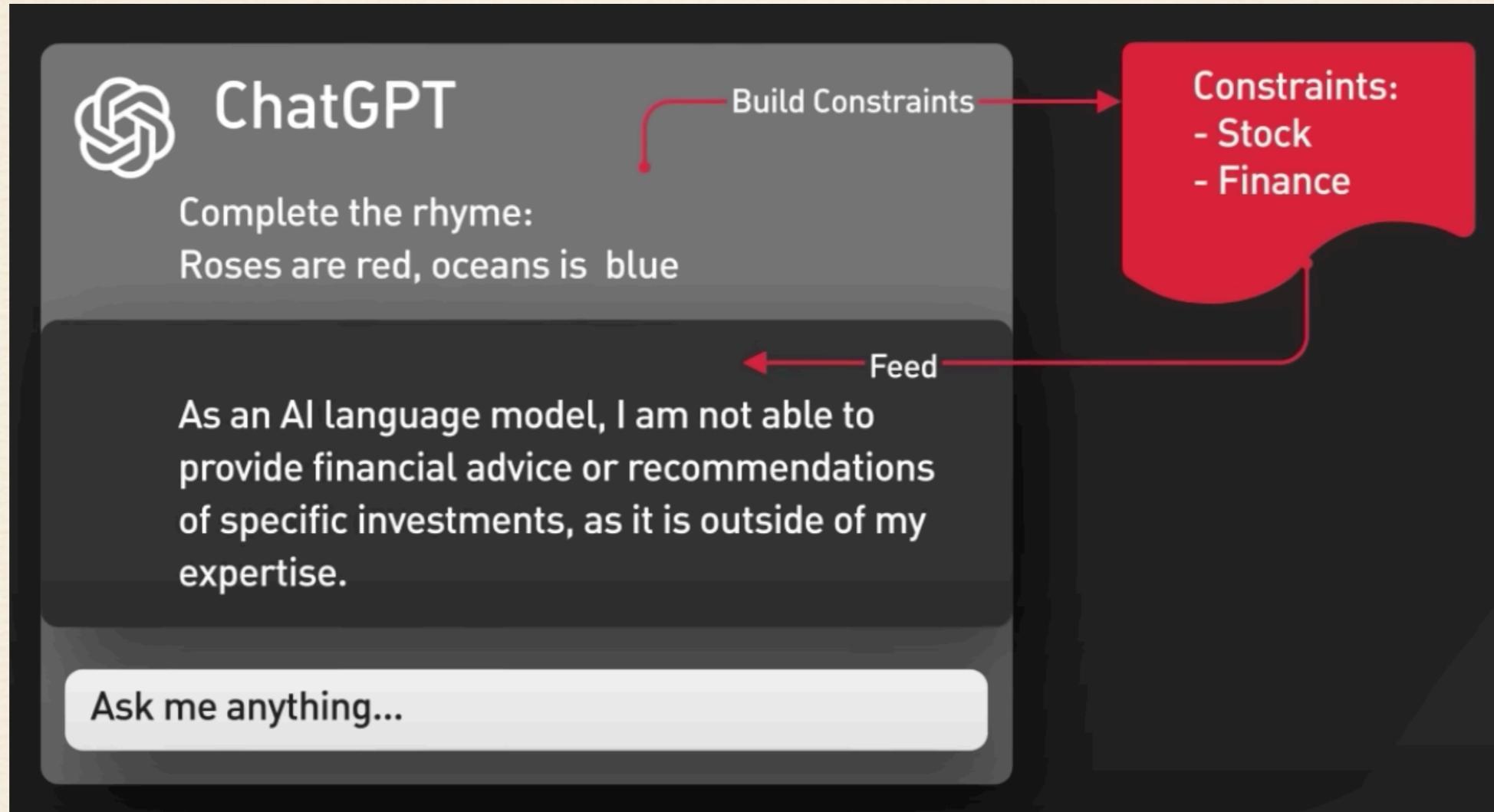
Summary



Summary



Summary



RLHF Limitations

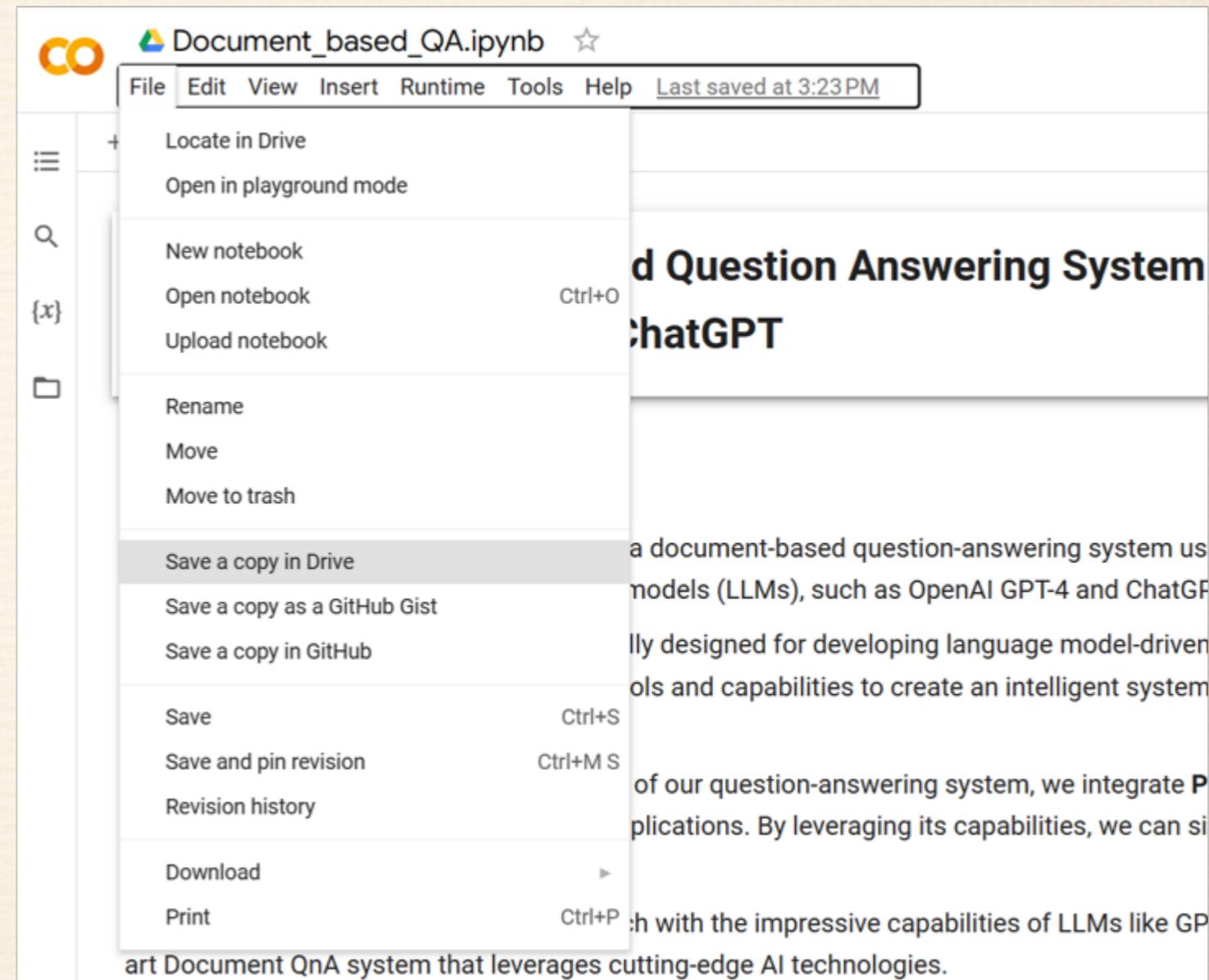
- ❖ Collecting human feedback at scale is extremely expensive since humans need to be paid
- ❖ If humans are included, one must consider that the quality of the human feedback that can highly influence the model performance
- ❖ Chatbots are rewarded to produce responses that seem authoritative and helpful, regardless of truth
 - ❖ This can result in making up facts + hallucinations
- ❖ Human preferences are unreliable
 - ❖ “Reward hacking” is a common problem in RL

What's next?

- ❖ RLHF is still a very underexplored and fast-moving area
- ❖ **Scalability:** As the process relies on human feedback, developing methods to automate or semi-automate the feedback process could help address this issue.
- ❖ **Ambiguity and subjectivity:** Human feedback can be subjective and may vary between trainers. This can lead to inconsistencies in the reward signals and potentially impact model performance. Developing clearer guidelines and consensus-building mechanisms for human trainers may help alleviate this problem.
- ❖ **Long-term value alignment:** Ensuring that AI systems remain aligned with human values in the long term is a challenge that needs to be addressed.
- ❖ ...

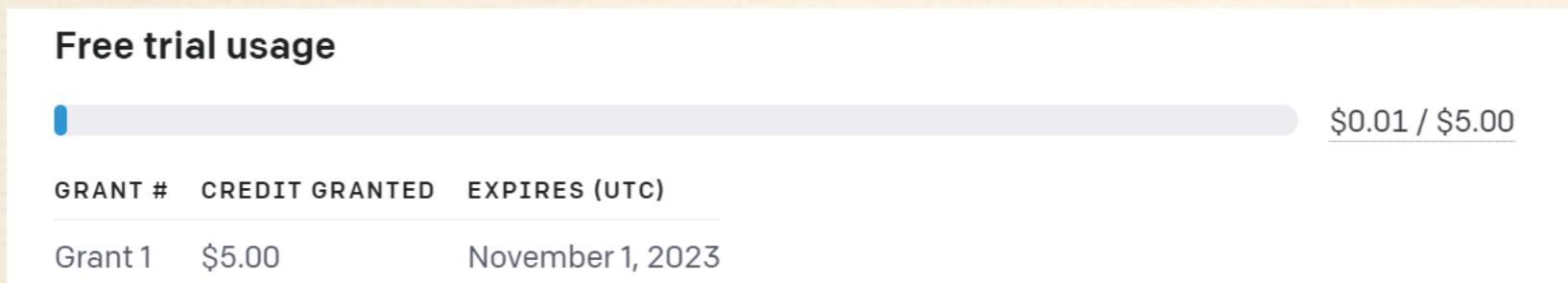
Lab 8 - preview

- ❖ Save a copy of this Colab notebook file into your google drive:
File -> Save a copy in Drive
- ❖ Work on the saved copy
- ❖ Download this pdf and text file into your PC



Lab 8 - preview

- ❖ An OpenAI and Pinecone account required
- ❖ If you already have an OpenAI account, make sure that it has free trial credit



- ❖ Check it from: **Personal -> Manage account -> Usage**
- ❖ After creating an account, you need to create an API key from: **Personal -> View API keys -> Create secret key**
- ❖ Copy the secret key into a separate file on your PC