

The “Deep Learning for NLP” Lecture Roadmap

Lecture 10: LLMs (2/2)

~~Lecture 5: Text Vectorization
and the Bag-of-Words Model~~

~~Lecture 6: Embeddings~~

~~Lecture 7: Transformers – (1/2)~~

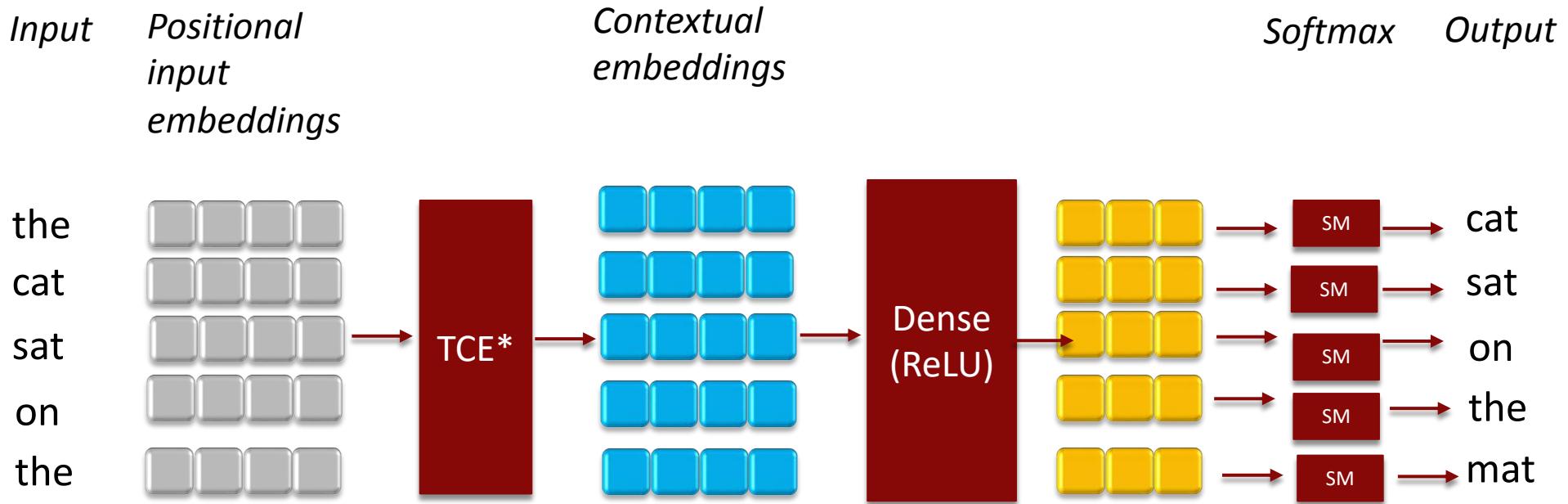
~~Lecture 8: Transformers – (2/2)~~

~~Lecture 9: LLMs (1/2)~~



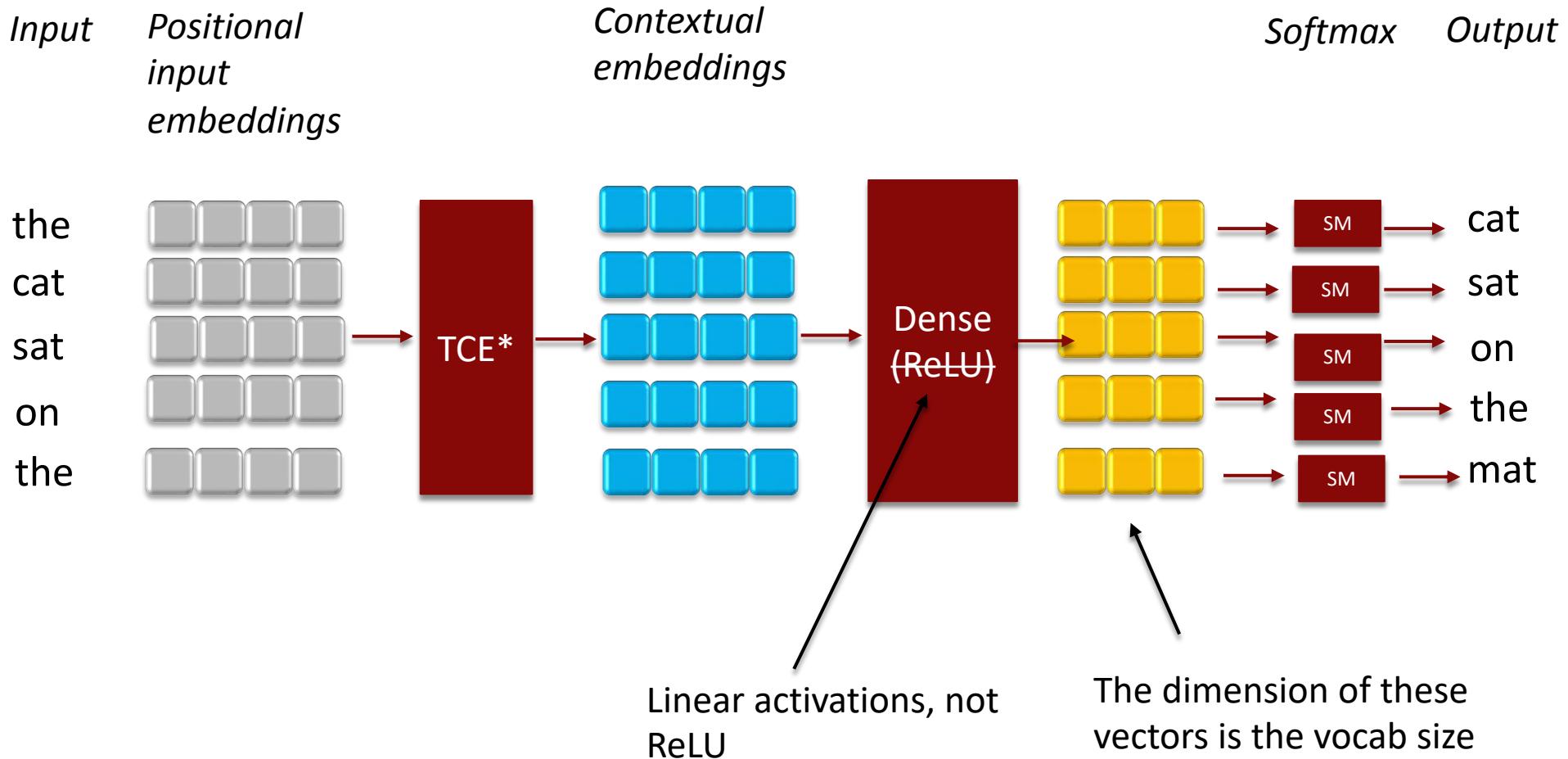
15.S04: Hands-on Deep Learning
Spring 2024
Farias, Ramakrishnan

In the last class, we saw how an autoregressive LLM like GPT-3 can be built for next-word-prediction using Transformer Causal Encoders



*Transformer **Causal** Encoder stack

Correction: The contextual embeddings flow into a Dense layer with **linear** activations (not ReLU) followed by a Softmax



*Transformer **Causal** Encoder stack

What about GPT and GPT-2?

- GPT stands for Generative Pretrained Transformer
- Like GPT-3, GPT and GPT-2 were trained in a very similar “predict the next word” fashion
- Two key differences
 - GPT-3 was trained on much more data
 - GPT-3’s underlying NN was much bigger

Model	Number of parameters
GPT	117 million
GPT-2	1.5 billion
GPT-3	175 billion

Unlike GPT and GPT-2, GPT-3 could generate amazing *completions* from a starting prompt



The importance of being on twitter

by Jerome K. Jerome
London, Summer 1897

It

<https://twitter.com/quasimondo/status/1284509525500989445> via
<https://www.technologyreview.com/2023/02/08/1068068/ChatGPT-is-everywhere-heres-where-it-came-from/>

Unlike GPT and GPT-2, GPT-3 could generate amazing *completions* from a starting prompt

Prompt

The importance of being on twitter

by Jerome K. Jerome
London, Summer 1897

It

Generated by GPT-3

is a curious fact that the last remaining form of social life in which the people of London are still interested is Twitter. I was struck with this curious fact when I went on one of my periodical holidays to the sea-side, and found the whole place twittering like a starling-cage. I called it an anomaly, and it is.

I spoke to the sexton, whose cottage, like all sexton's cottages, is full of antiquities and interesting relics of former centuries. I said to him, "My dear sexton, what does all this twittering mean?" And he replied, "Why, sir, of course it means Twitter." "Ah!" I said, "I know about that. But what is Twitter?"

"It is a system of short and pithy sentences strung together in groups, for the purpose of conveying useful information to the initiated, and entertainment and the exercise of wits to the initiated, and entertainment and the exercise of wits to the rest of us."



But it was not good at following user
instructions



Let's make this request* to GPT-3:

help me write a short note to introduce myself to my neighbor

GPT-3's response



help me write a short note to introduce myself to my neighbor.

what is a good introduction to a resume.

good introduction to a resume.

best photos of good resume introduction example sample.

GPT-3's response

help me write a short note to introduce myself to my neighbor.

what is a good introduction to a resume?



good introduction to a resume.

best photos of good resume introduction example sample.



In addition to unhelpful answers, it can produce offensive or factually incorrect output as well

Why GPT-3 sometimes produced unhelpful answers



- GPT-3 wasn't *explicitly* trained to generate good answers to user instructions.



Why GPT-3 sometimes produced unhelpful answers



- GPT-3 wasn't explicitly trained to generate good answers to user instructions.
 - It is simply trying to guess the next word repeatedly, using what it has gleaned from the billions of sentences it was trained on.
-



Let's *explicitly* train it to follow instructions!

OpenAI developed an approach called **Instruction Tuning**

Training language models to follow instructions with human feedback

Long Ouyang* **Jeff Wu*** **Xu Jiang*** **Diogo Almeida*** **Carroll L. Wainwright***

Pamela Mishkin* **Chong Zhang** **Sandhini Agarwal** **Katarina Slama** **Alex Ray**

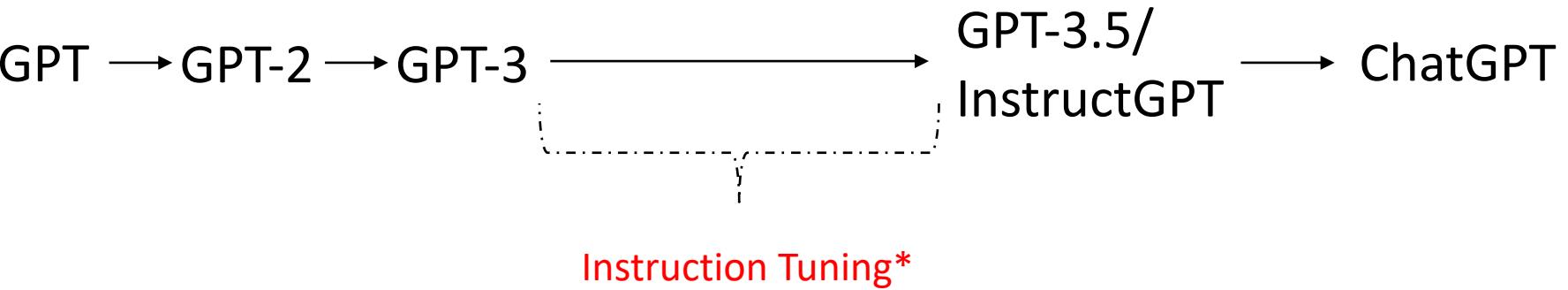
John Schulman **Jacob Hilton** **Fraser Kelton** **Luke Miller** **Maddie Simens**

Amanda Askell[†] **Peter Welinder** **Paul Christiano*[†]**

Jan Leike* **Ryan Lowe***

OpenAI

From GPT-3 to GPT-3.5



*The approach has two main steps: (1) Supervised Fine-Tuning (2) Reinforcement Learning from Human Feedback.
“Instruction Tuning” usually refers to just (1) but we will use it as a shorthand to refer to the full approach. See
<http://arxiv.org/abs/2203.02155> for details.

Instruction Tuning – Step 1



- Get humans to write (high-quality) answers to instructions. About 12,500 such instruction-answer pairs were created.

Instruction Tuning – Step 1

- Get humans to write (high-quality) answers to instructions. About 12,500 such instruction-answer pairs were created. Example:

Instruction	GPT-3 Answer	Human-created answer
<i>Explain the moon landing to a 6 year old in a few sentences.</i>	Explain the theory of gravity to a 6 year old.	People went to the moon in a big rocket, walked around and came back to Earth. They took pictures of what they saw and sent them back so we could all see them.*

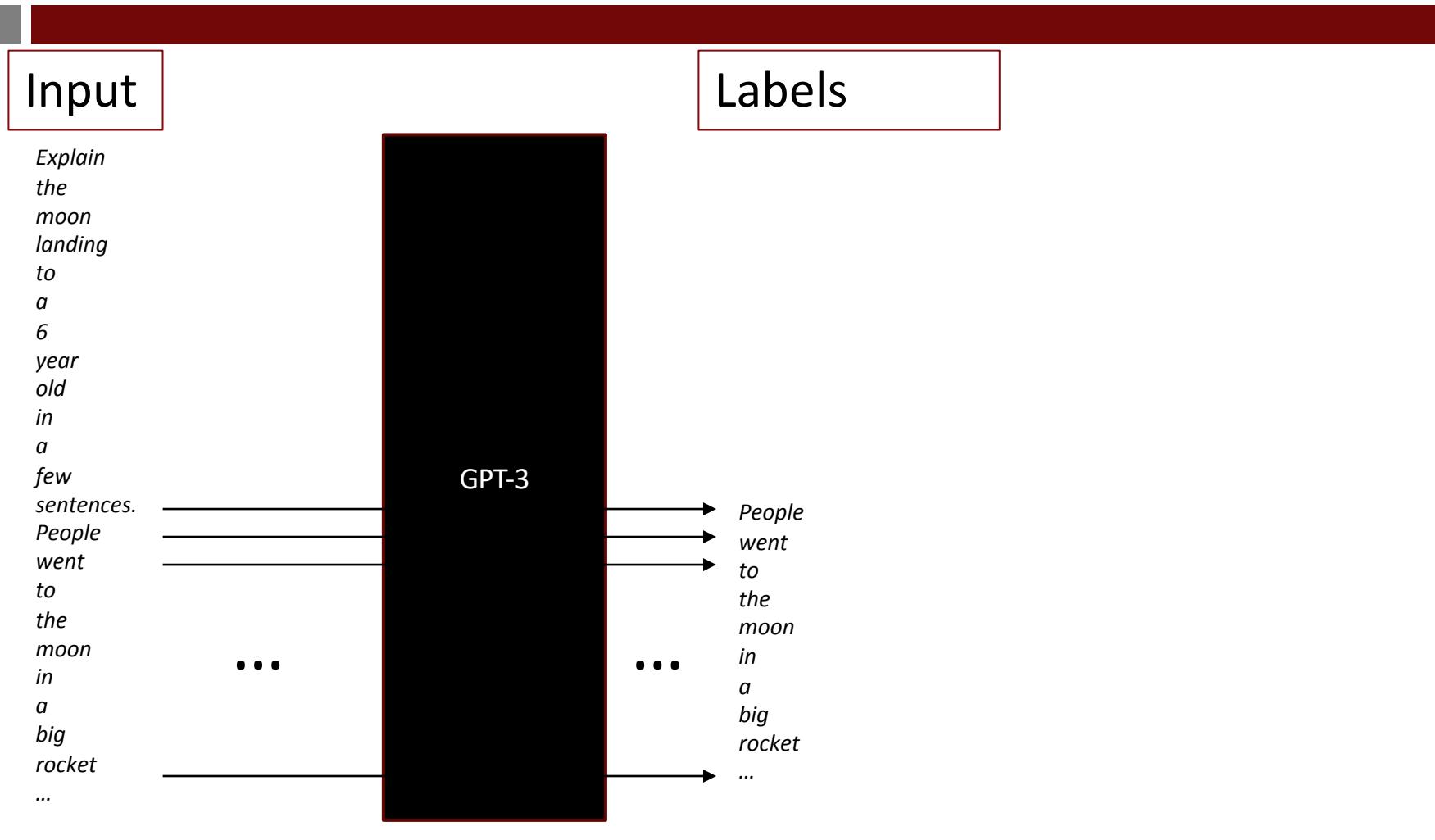
*Lightly edited version of what's in <https://openai.com/blog/instruction-following/>

Instruction Tuning – Step 1



- Get humans to write (high-quality) answers to instructions. About 12,500 such instruction-answer pairs were created.
 - Using these ~12,500 question-answer pairs as training data, train GPT-3 some more using next-word-prediction (next slide)
-

Instruction Tuning – Step 1



This step – called “Supervised Fine Tuning” (SFT) - helped

- GPT-3 did much better on instructions. We would like to do more SFT.



This step – called “Supervised Fine Tuning” (SFT) - helped

- GPT-3 did much better on instructions. We would like to do more SFT.
- *But writing high-quality answers to thousands of instructions is difficult and expensive*



What's easier than writing a good answer?



What's easier than writing a good answer?



- *Ranking answers written by somebody else!*
-

What's easier than writing a good answer?



- *Ranking answers written by somebody else!*
 - We can ask GPT-3 to generate several answers to a question ...
-

What's easier than writing a good answer?

- *Ranking answers written by somebody else!*
- We can ask GPT-3 to generate several answers to a question ...

How?

If we ask GPT-3 to sample the next word, remember that it can generate several next-words for the same input.

What's easier than writing a good answer?



- *Ranking answers written by somebody else!*
 - We can ask GPT-3 to generate several answers to a question ...
 - ... and have humans simply rank them from most useful to least useful!
-

Instruction Tuning – Step 2



- Open AI collected 33,000 instructions, fed them to GPT-3 and generated several answers to each instruction, and *had humans simply rank the answers for each instruction from most helpful to least helpful*

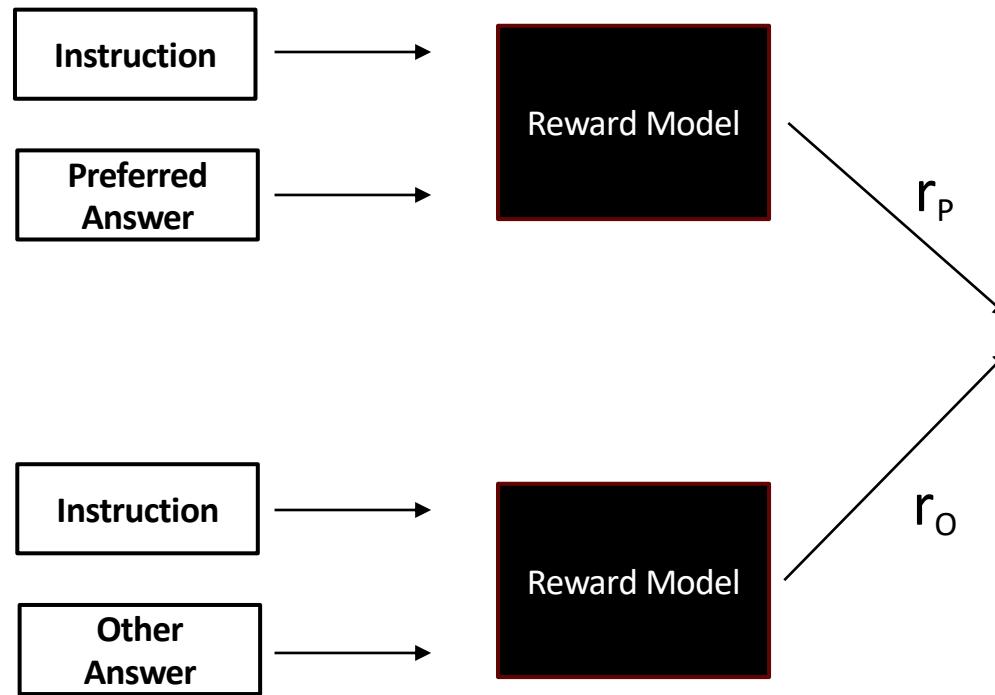
Instruction Tuning – Step 2



- Open AI collected 33,000 instructions, fed them to GPT-3 and generated several answers to each instruction, and had humans simply rank the answers for each instruction from most helpful to least helpful
- Using this data, a training dataset was assembled
 - Given an Instruction and two answers A and B, let's say the human ranks A as better than B. The resulting data point is [Instruction, Preferred answer = A, Other answer = B]
- and a Reward model was trained*

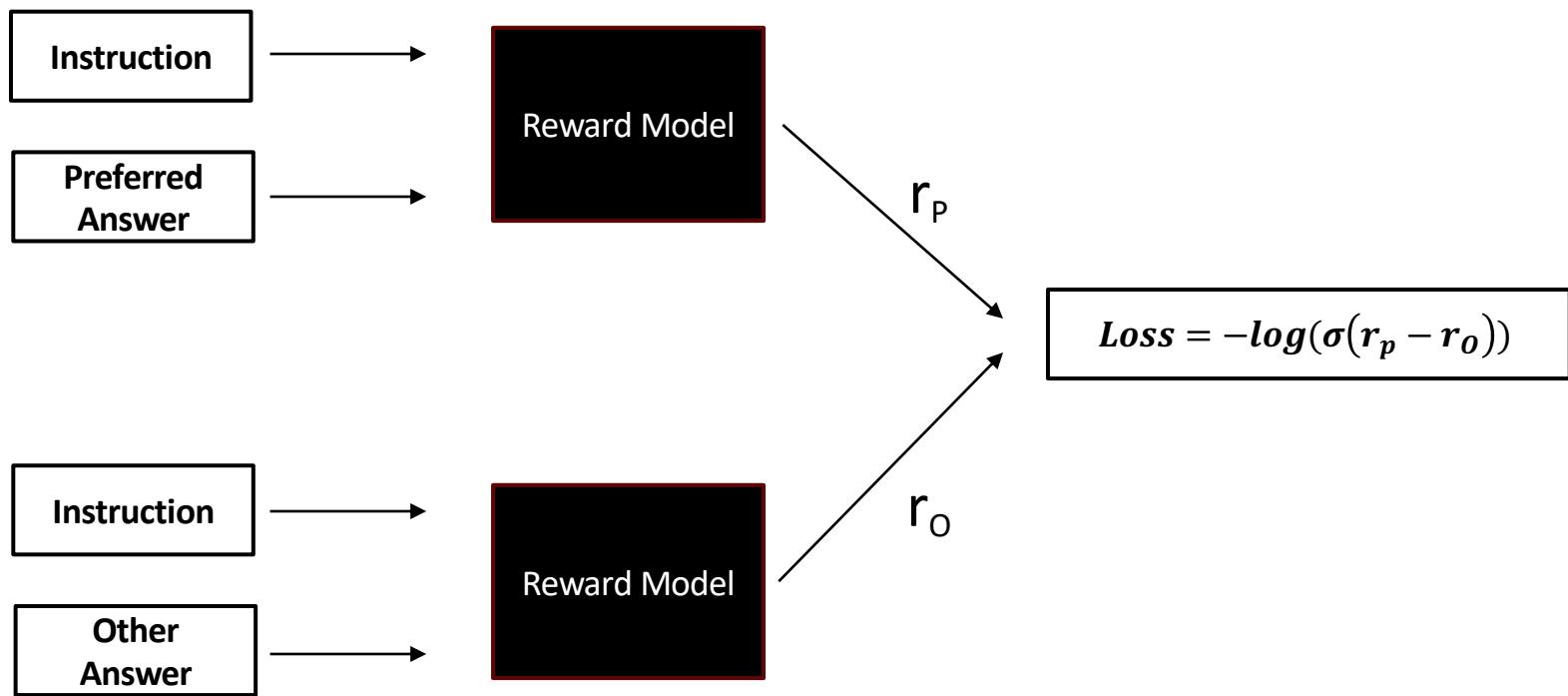
*For ease of exposition, we are describing the slightly simpler version in <http://arxiv.org/abs/2009.01325>

The Reward Model has a single numerical output. We want this number to indicate how good the answer is for that particular instruction

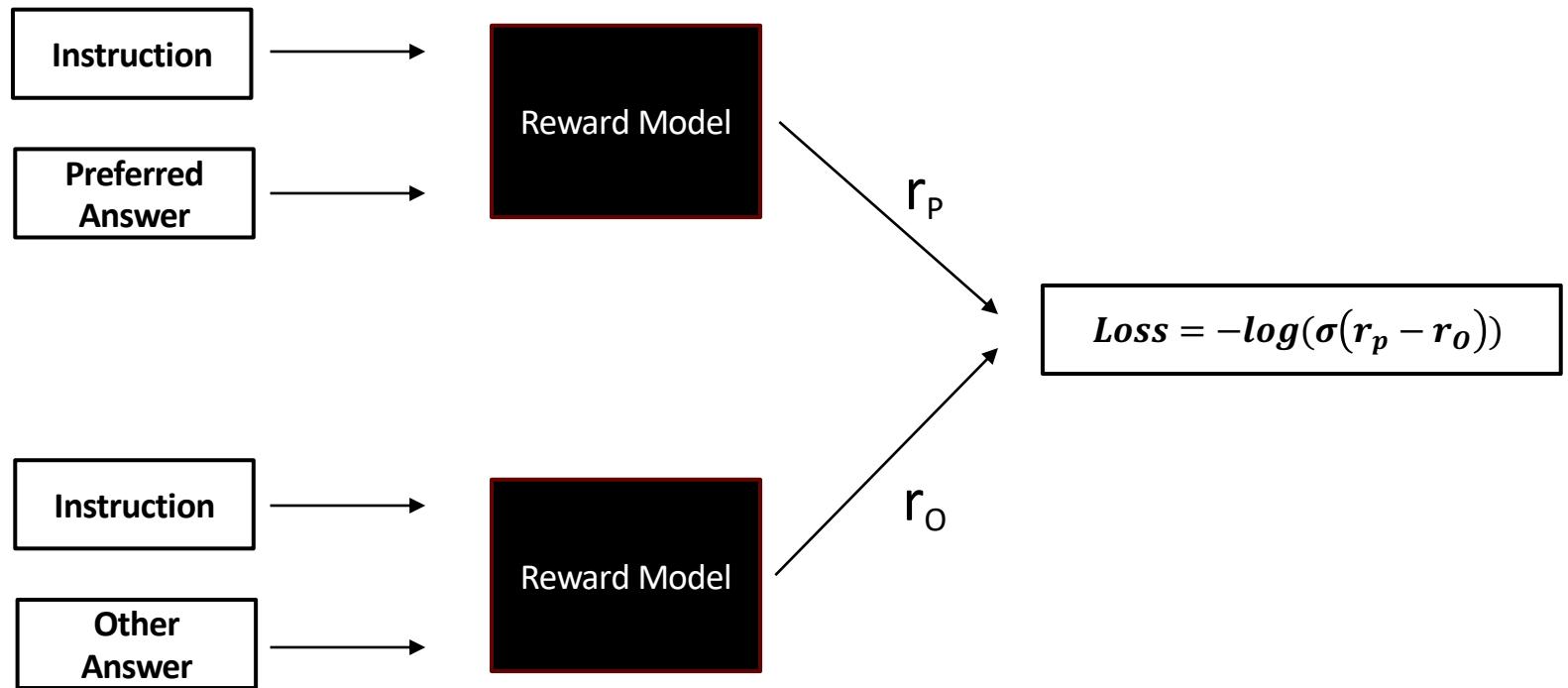


The two models are exact copies

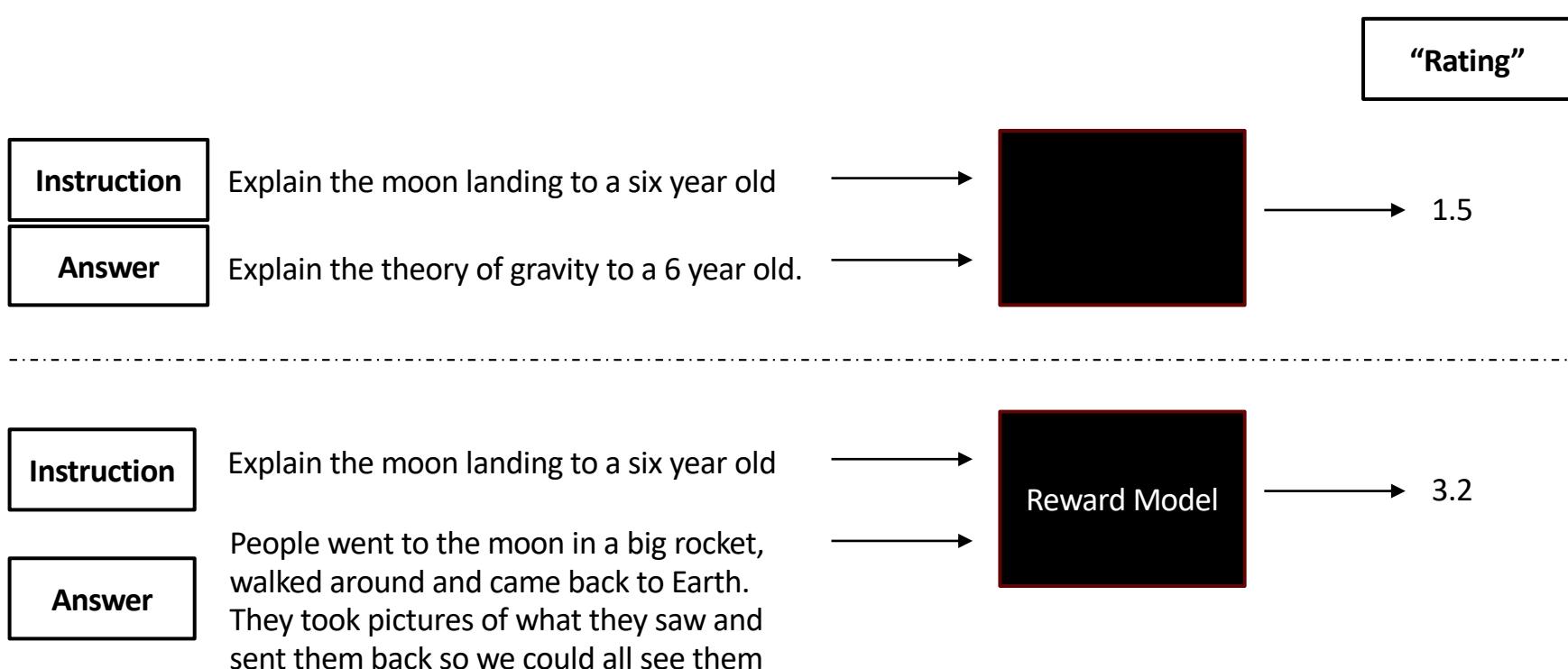
We want the model to give a lower rating for the Other answer compared to the Preferred answer, so we define a loss function that encourages this ...



... and train this model using SGD (or variants)



After training, Reward Model can provide a numerical *rating* for *any* [instruction, answer] pair





In essence, the Reward Model has learned how humans rank responses.

This can be used to improve GPT-3 further.

We send an instruction into GPT-3 and get an answer



Instruction

Answer

Explain the moon
landing to a six
year old

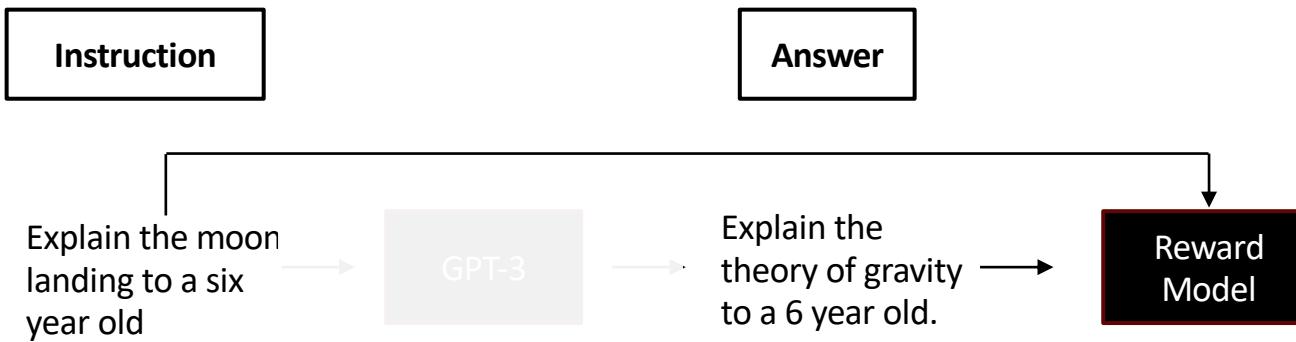


GPT-3

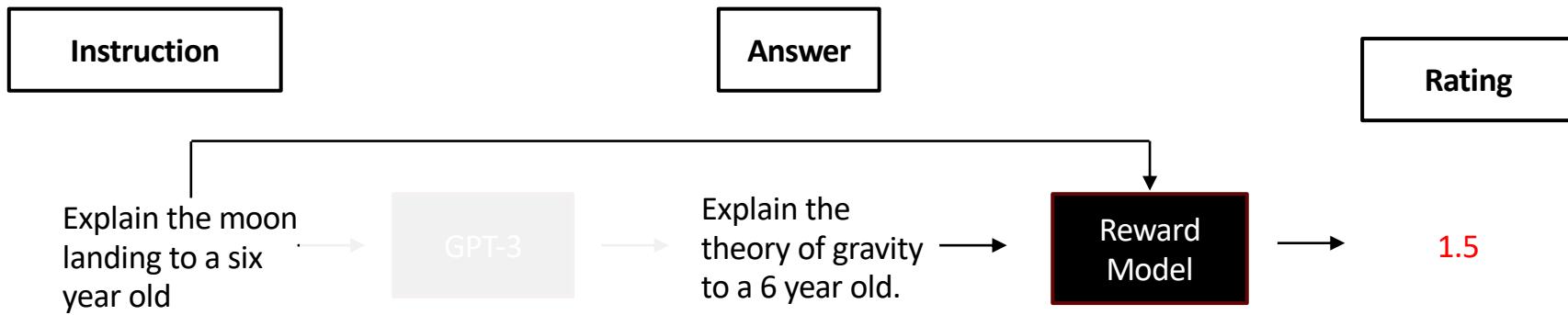


Explain the
theory of gravity
to a 6 year old.

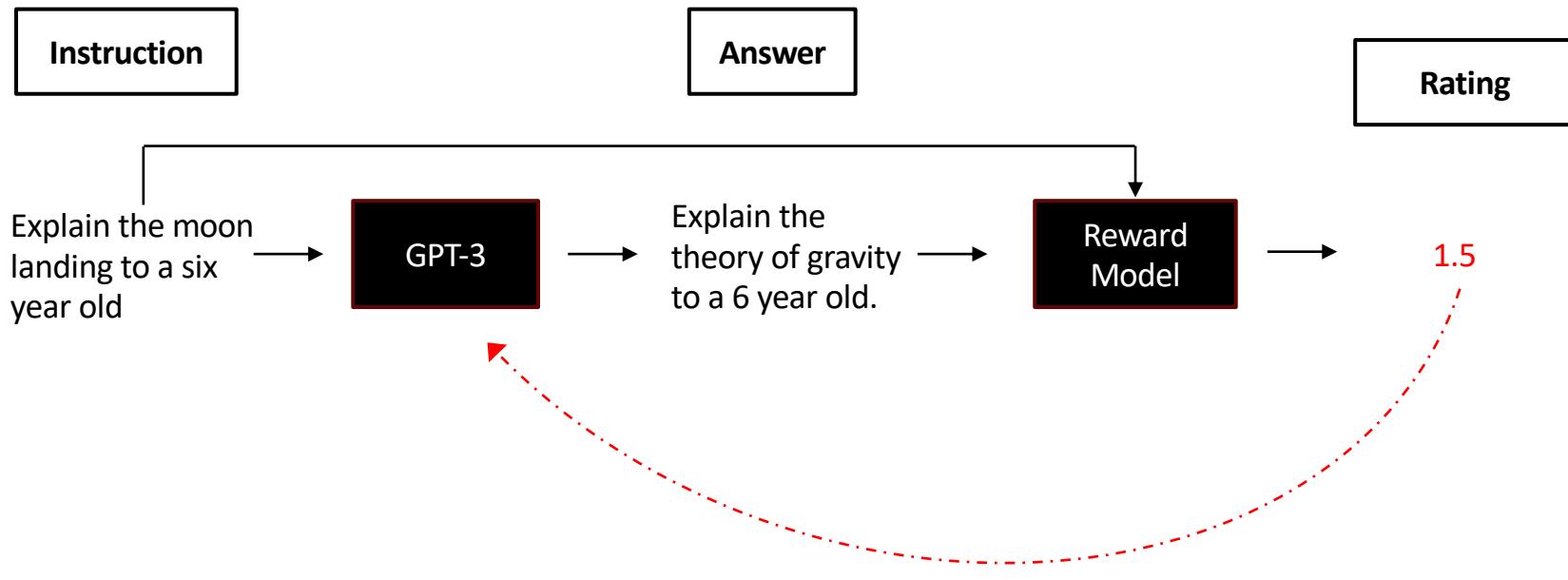
Next, we feed the question *and* the answer to the Reward Model



Next, we feed the question *and* the answer to the Reward Model **which outputs a rating**

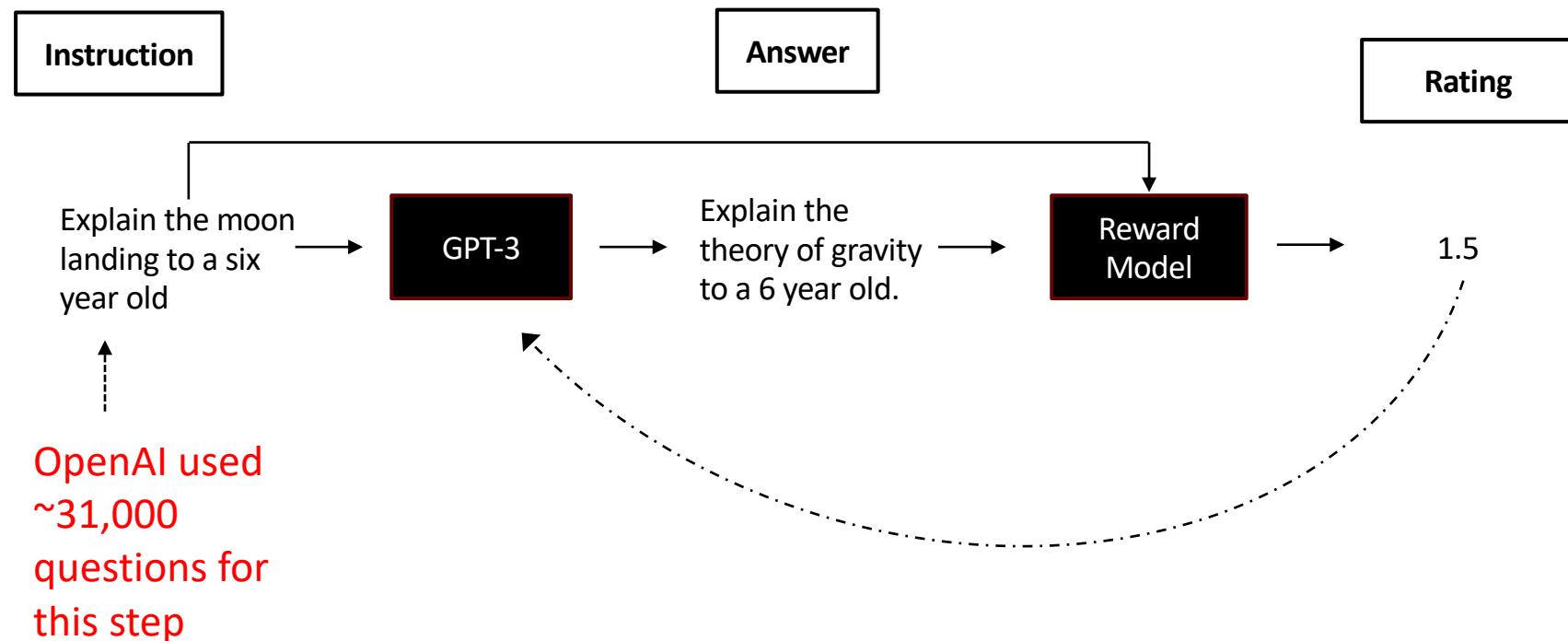


We use this rating to “nudge”* GPT-3 in the right direction



* Using a [Reinforcement Learning](#) algorithm called [Proximal Policy Optimization](#)

This is repeated **thousands** of times with instruction-answer pairs



That's how GPT-3 became GPT-3.5/ InstructGPT



- Fine-tune GPT-3 using human-written question-answer pairs.
- Have GPT-3 provide multiple answers for each question and have humans rank the answers. With this training data, build the Reward Model
- Using the rating from the Reward Model, fine-tune GPT3 using Reinforcement Learning

BTW, Steps 2 and 3 are referred to as “Reinforcement Learning from Human Feedback”

- Fine-tune GPT-3 using human-written question-answer pairs.
- Have GPT-3 provide multiple answers for each question and **have humans rank the answers**. With this training data, build the Reward Model
- Using the rating from the Reward Model, fine-tune GPT3 using Reinforcement Learning

BTW, you are providing training data for the Reward Model when you do this ...

RA

explain the difference between polysemy and synonymy



Polysemy and synonymy are two concepts that are often used in linguistics to describe the relationships between words.

Polysemy refers to the phenomenon where a single word has multiple related meanings, while synonymy refers to the phenomenon where two or more words have the same or nearly the same meaning.



OK, let's see how GPT 3.5/InstructGPT does on this prompt



help me write a short note to introduce myself to my neighbor.



OK, let's see how GPT 3.5/InstructGPT does on this prompt

help me write a short note to introduce myself to my neighbor.



Hello Neighbor,

My name is [Name] and I recently moved into the neighborhood. I am excited to get to know my new neighbors and am looking forward to getting to know you. If you ever need anything, please don't hesitate to reach out.

Best,
[Name]

Clearly better than GPT-3!

help me write a short note to introduce myself to my neighbor.



Hello Neighbor,

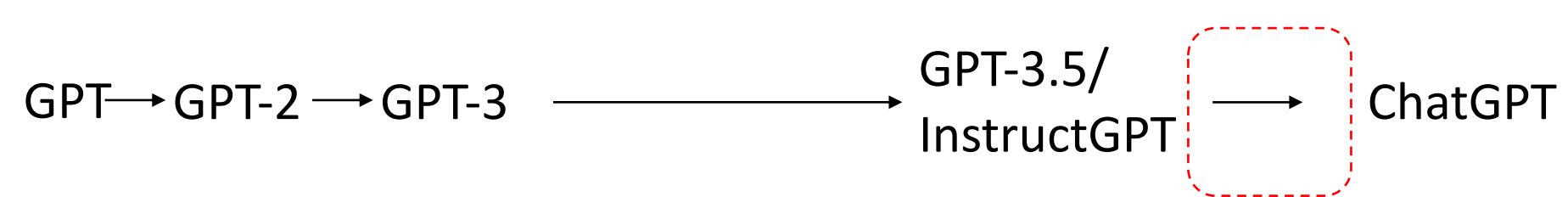
My name is [Name] and I recently moved into the neighborhood. I am excited to get to know my new neighbors and am looking forward to getting to know you. If you ever need anything, please don't hesitate to reach out.



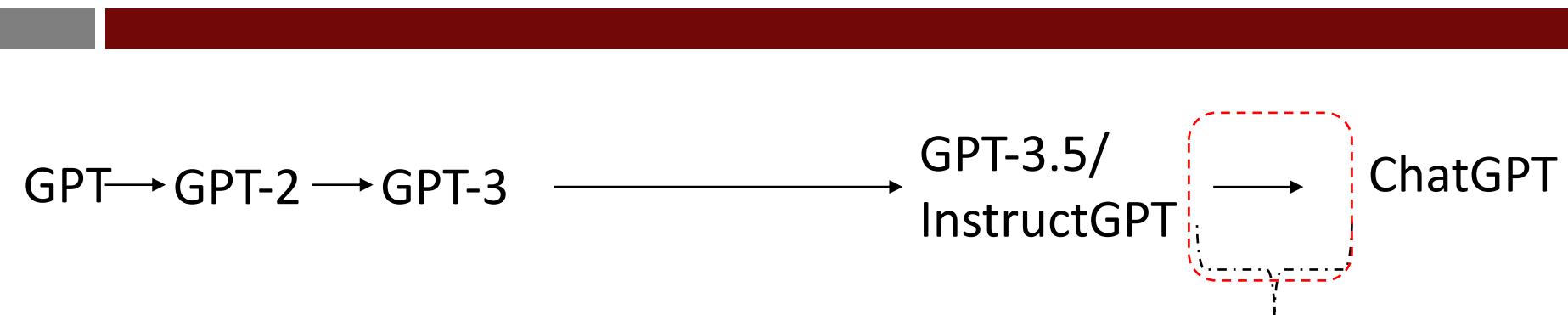
Best,

[Name]

From GPT-3.5 to ChatGPT



From GPT-3.5 to ChatGPT



This is very similar to the 3-step process we followed to get from GPT-3 to GPT-3.5/InstructGPT but with a key difference

Instead of just instruction-answer pairs, use conversations as the training data

With this conversational dataset, the same process was followed – Supervised Fine-tuning, Reward Model, Reinforcement Learning etc.

ChatGPT does much better on our prompt

RA

help me write a short note to introduce myself to my neighbor.



Dear [Neighbor's Name],

I hope this note finds you well. My name is [Your Name], and I am your new neighbor. I recently moved into the [apartment/house] next to yours and wanted to take a moment to introduce myself.

I am a [profession/hobby] and enjoy spending my free time [activity/interest]. I also love meeting new people and getting to know my community. If you ever need anything or would like to chat, please don't hesitate to stop by and say hello.

I look forward to getting to know you and being a part of this wonderful neighborhood.

Best regards,
[Your Name]



But because ChatGPT is trained on conversations , we can ask follow-on questions

RA

help me write a short note to introduce myself to my neighbor.



Dear [Neighbor's Name],

I hope this note finds you well. My name is [Your Name], and I am your new neighbor. I recently moved into the [apartment/house] next to yours and wanted to take a moment to introduce myself.

I am a [profession/hobby] and enjoy spending my free time [activity/interest]. I also love meeting new people and getting to know my community. If you ever need anything or would like to chat, please don't hesitate to stop by and say hello.

I look forward to getting to know you and being a part of this wonderful neighborhood.

Best regards,
[Your Name]

But because ChatGPT is trained on conversations , we can ask follow-on questions

RA

help me write a short note to introduce myself to my neighbor.



Dear [Neighbor's Name],

I hope this note finds you well. My name is [Your Name], and I am your new neighbor. I recently moved into the [apartment/house] next to yours and wanted to take a moment to introduce myself.

I am a [profession/hobby] and enjoy spending my free time [activity/interest]. I also love meeting new people and getting to know my community. If you ever need anything or would like to chat, please don't hesitate to stop by and say hello.

I look forward to getting to know you and being a part of this wonderful neighborhood.

Best regards,
[Your Name]

RA

can you make it more formal?



Certainly, here's a more formal version:

Dear [Neighbor's Name],

I hope this letter finds you in good health and spirits. My name is [Your Name], and I am your new neighbor. I recently moved into the [apartment/house] adjacent to yours and wanted to take a moment to introduce myself formally.

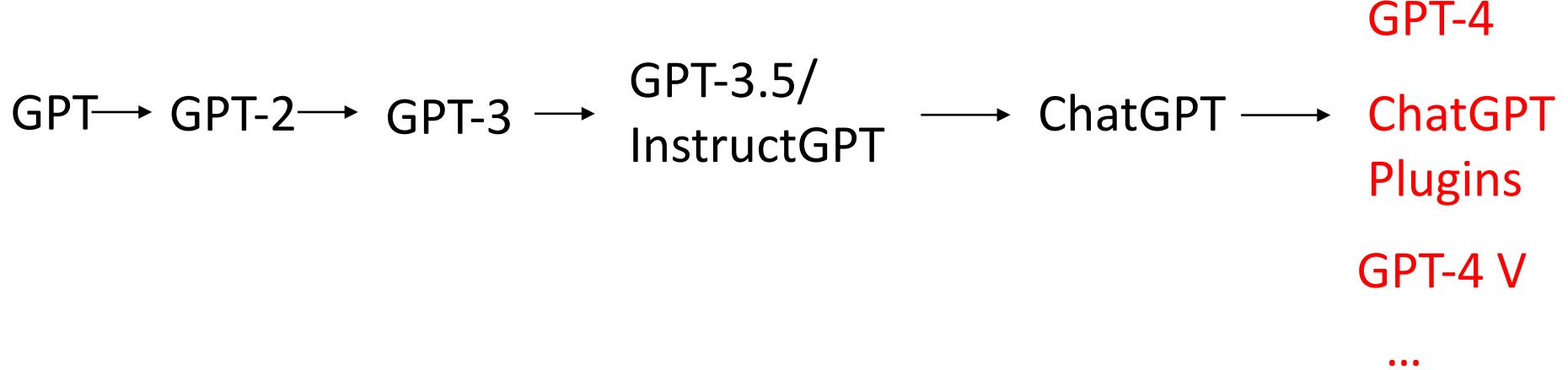
As a [profession/hobby], I value building strong relationships with my neighbors and am excited to get to know the community. In my free time, I enjoy [activity/interest], and I am always open to discovering new hobbies and interests.

I wanted to extend an invitation to stop by anytime if you need any assistance or if you would like to have a friendly conversation. I look forward to getting to know you and being a part of this wonderful neighborhood.

Thank you for taking the time to read my letter. Please accept my warmest regards.

Sincerely,

The journey continues ...



This is Instruction Tuning in a nutshell

Step 1

Start with a base LLM (like GPT-3) that has been trained to predict the next word.

Step 2

Get humans to write responses to thousands of instructions. Fine-tune the base LLM with this data.

Step 3

Generate multiple responses to each instruction using the fine-tuned LLM. Ask humans to rank the responses. Do this for several thousand instructions.

Step 4

Use this data to build a Reward Model.

Step 5

Using the Reward Model, train the fine-tuned LLM further using Reinforcement Learning with several thousand questions.

As you saw, significant **human** effort was involved in Steps 2 and 3

Step 1

Start with a base LLM (like GPT-3) that has been trained to predict the next word.

Step 2

Get **humans** to write responses to **thousands** of instructions. Fine-tune the base LLM with this data.

Step 3

Generate multiple responses to each instruction using the fine-tuned LLM. Ask **humans** to rank the responses. Do this for **several thousand** instructions.

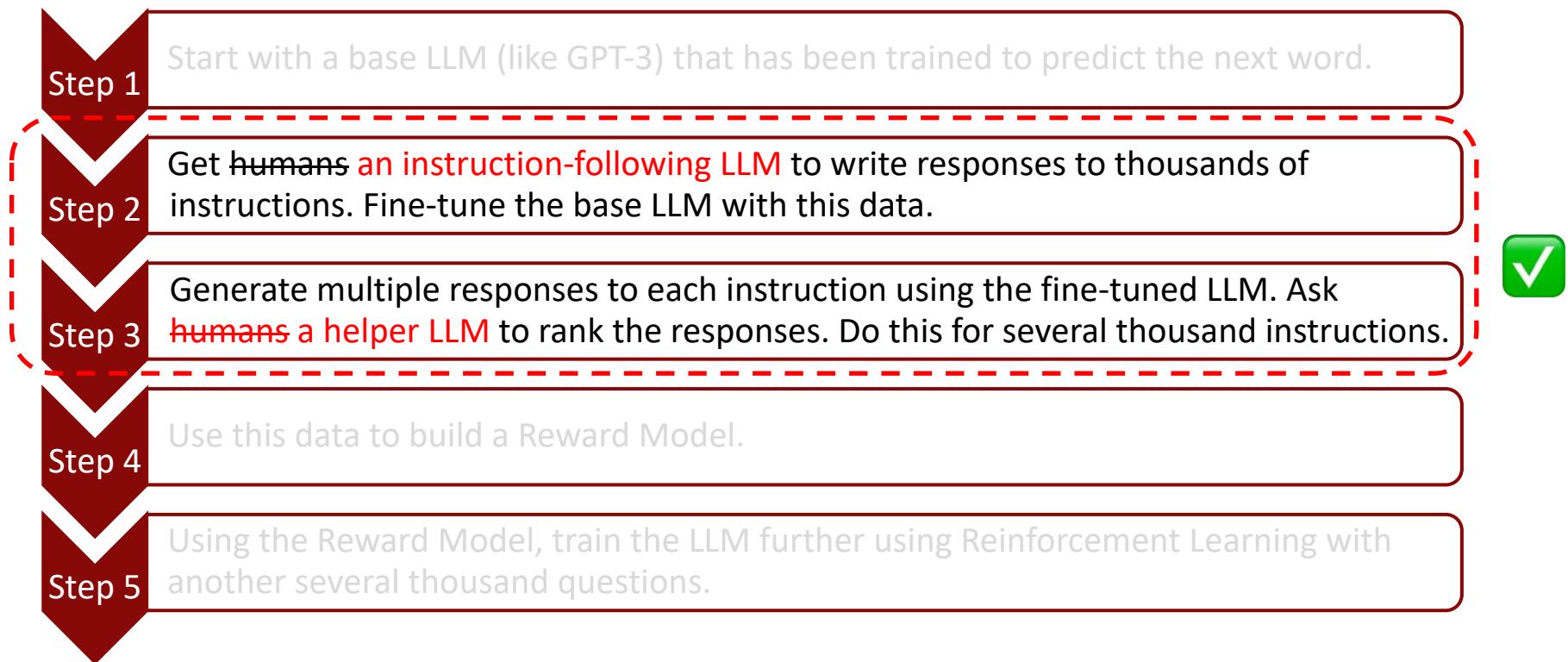
Step 4

Use this data to build a Reward Model.

Step 5

Using the Reward Model, train the fine-tuned LLM further using Reinforcement Learning with another thousand questions.

Using “helper LLMs”, researchers have automated these steps as well!



If you are curious to learn more...



Rama Ramakrishnan (He/Him) • You

Professor of the Practice at MIT Sloan School of Management

1mo •

...

Over the past year, researchers have invented clever ways to use LLMs as "helpers" to create other LLMs cheaper and faster. It has made me realize just how versatile LLMs can be and I wanted to share what I have learned. Please see in full-screen mode. Link to PDF below. **#chatgpt #gpt4 #genai**

How to use LLMs as “helpers” to build and customize other LLMs

A quick non-technical intro

*Prof. Rama Ramakrishnan
MIT Sloan School of Management
January 10, 2023*

<https://www.linkedin.com/feed/update/urn:li:activity:7150937271251136514/>

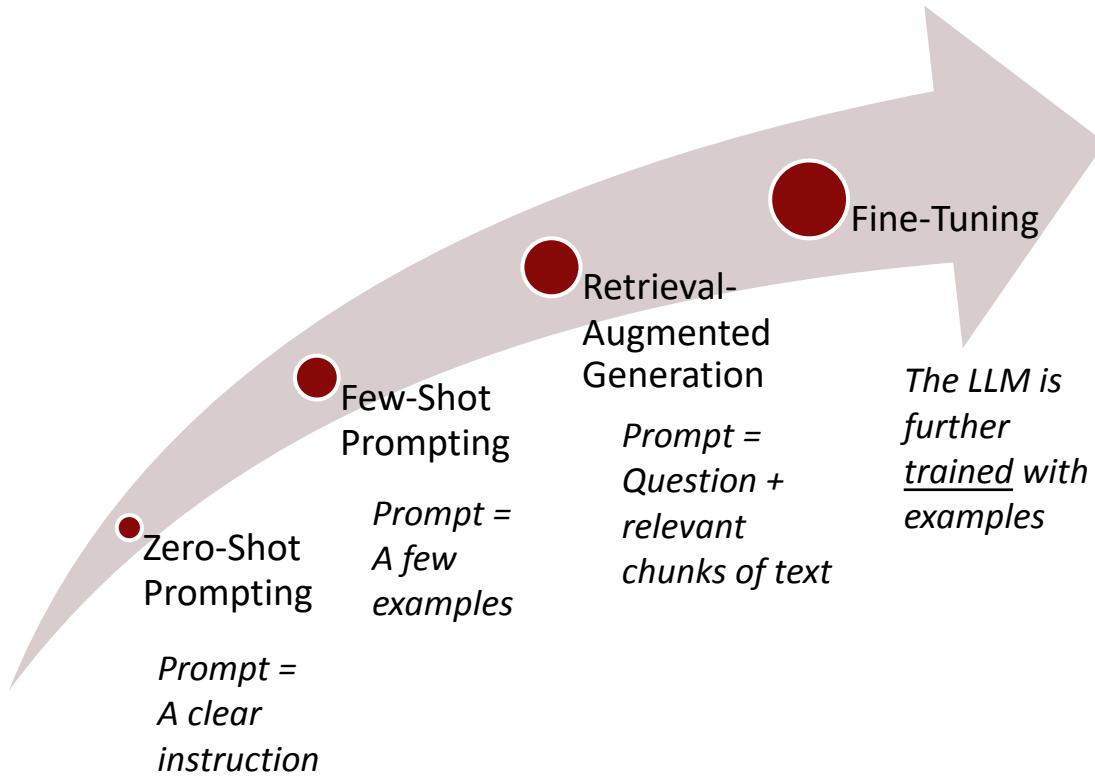
The importance of adapting base LLMs

- To make a base LLM like GPT-3 useful at understanding language and responding to instructions, we had to **adapt** it with high-quality data (using Supervised Fine-Tuning with and Reinforcement Learning with Human Feedback (RLHF))
- This holds true more generally. To make a base LLM useful for specific business applications, we may need to adapt them with business/domain-specific data. In the next section, we will look at techniques for doing so.



“Adaptation” is the process of taking
a base (aka foundation or base) LLM
and tailoring it for a specific task

The Ladder of LLM Adaptation



Example of zero-shot prompting: We want to build a product review defect detector



★ ★ ★ ★ ★

Frame Finish: Black

The curve of the back of the chair does not leave enough room to sit comfortably

© Wayfair LLC. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.

<https://www.wayfair.com/furniture/pdp/latitude-run-alori-task-chair-w005270016.html>

We can directly “instruct” the LLM to check if the review describes a defect

The screenshot shows the Hugging Face Playground interface. At the top, there is a navigation bar with icons for Overview, Documentation, Examples, and Playground, where 'Playground' is highlighted in green. Below the navigation bar, the word 'Playground' is displayed in bold black text. A red arrow points from the left towards the text input area, labeled 'Prompt →'. In the text input area, there is a dashed red box containing the following text:
Tell me if a product defect is being described in this product review:
The curve of the back of the chair does not leave enough room to sit comfortably

The LLM gets it right!



Overview

Documentation

Examples

Playground

Playground

Prompt →

Tell me if a product defect is being described in this product review:

The curve of the back of the chair does not leave enough room to sit comfortably

Response →

>Yes, this is a product defect.



When adapting an LLM to a specific task using zero-shot, we may need to carefully design the prompt so that the LLM gives you the desired response. This is called “prompt engineering”

Example: A simple question that ChatGPT sometimes gets wrong

No prompt engineering

RA

what is the fifth word of this sentence?



The fifth word of this sentence is "is."

(HT: Twitter user @deanbuono)

With a little prompt engineering, it can answer the question reliably

No prompt engineering

RA

What is the fifth word of this sentence?



The fifth word of the sentence is "the." 

A “little” prompt engineering

RA

i will give you a sentence. First, list all the words that are in the sentence. Then tell me the fifth word. Sentence: what is the fifth word of this sentence?



The words in the sentence are: "what," "is," "the," "fifth," "word," "of," "this," "sentence."

The fifth word of the sentence is "word." 

Researchers are “discovering” some simple and widely-applicable prompt engineering strategies

“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: The answer (arabic numerals) is

(Output) 8 ✘

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. ✓

Here's a recent "discovery"



*"Take a deep breath and work on
this problem step by step"*

Table 4: Test accuracies on GSM8K. We show the instruction with the highest test accuracy for each scorer-optimizer pair.

Scorer	Optimizer / Source	Instruction position	Top instruction	Acc
<i>Baselines</i>				
PaLM 2-L	(Kojima et al., 2022)	A_begin	Let's think step by step.	71.8
PaLM 2-L	(Zhou et al., 2022b)	A_begin	Let's work this out in a step by step way to be sure we have the right answer.	58.8
PaLM 2-L		A_begin	Let's solve the problem.	60.8
PaLM 2-L		A_begin	(empty string)	34.0
text-bison	(Kojima et al., 2022)	Q_begin	Let's think step by step.	64.4
text-bison	(Zhou et al., 2022b)	Q_begin	Let's work this out in a step by step way to be sure we have the right answer.	65.6
text-bison		Q_begin	Let's solve the problem.	59.1
text-bison		Q_begin	(empty string)	56.8
<i>Ours</i>				
PaLM 2-L	PaLM 2-L-IT	A_begin	Take a deep breath and work on this problem step-by-step.	80.2
PaLM 2-L	PaLM 2-L	A_begin	Break this down.	79.9
PaLM 2-L	gpt-3.5-turbo	A_begin	A little bit of arithmetic and a logical approach will help us quickly arrive at the solution to this problem.	78.5
PaLM 2-L	gpt-4	A_begin	Let's combine our numerical command and clear thinking to quickly and accurately decipher the answer.	74.5
text-bison	PaLM 2-L-IT	Q_begin	Let's work together to solve math word problems! First, we will read and discuss the problem together to make sure we understand it. Then, we will work together to find the solution. I will give you hints and help you work through the problem if you get stuck.	64.4
text-bison	text-bison	Q_end	Let's work through this problem step-by-step:	68.5
text-bison	gpt-3.5-turbo	Q_end	Analyze the given information, break down the problem into manageable steps, apply suitable mathematical operations, and provide a clear, accurate, and concise solution, ensuring precise rounding if necessary. Consider all variables and carefully consider the problem's context for an efficient solution.	66.5
text-bison	gpt-4	Q_begin	Start by dissecting the problem to highlight important numbers and their relations. Decide on the necessary mathematical operations like addition, subtraction, multiplication, or division, required for resolution. Implement these operations, keeping in mind any units or conditions. Round off by ensuring your solution fits the context of the problem to ensure accuracy.	62.7

There are many “prompt engineering” resources online. Here’s a recent one.

PROMPT DESIGN AND ENGINEERING: INTRODUCTION AND ADVANCED METHODS

Xavier Amatriain
xavier@amatriain.net

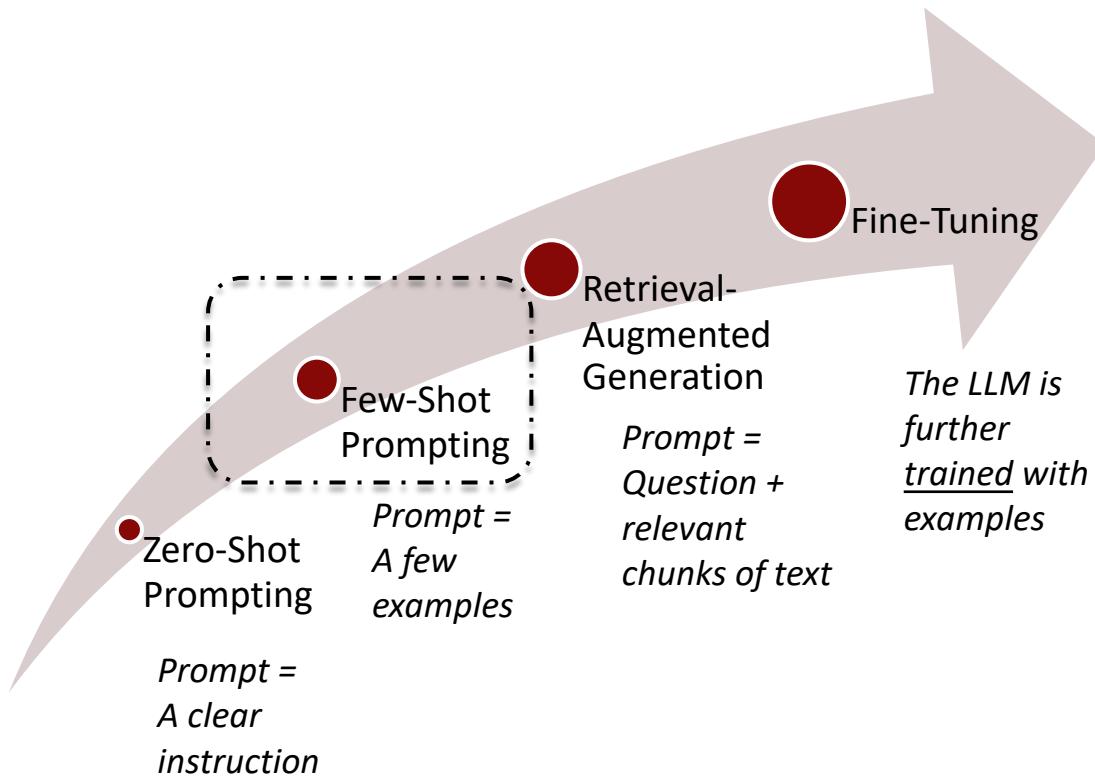
February 12, 2024

ABSTRACT

Prompt design and engineering has rapidly become essential for maximizing the potential of large language models. In this paper, we introduce core concepts, advanced techniques like Chain-of-Thought and Reflection, and the principles behind building LLM-based agents. Finally, we provide a survey of tools for prompt engineers.

<https://arxiv.org/abs/2401.14423>

Let's look at Few-Shot Prompting next



Let's say we want to build a “grammar corrector”



We collect a few examples of what we want the LLM to do

Prompt

Poor English input: I eated the purple berries.

Good English output: I ate the purple berries.

Poor English input: Thank you for picking me as your designer. I'd appreciate it.

Good English output: Thank you for choosing me as your designer. I appreciate it.

Poor English input: The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.

Good English output: The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.

Poor English input: I'd be more than happy to work with you in another project.

Prompt examples © Tom B. Brown, et al./ArXiv. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.

Notice the Good English – Poor English pattern

Prompt

Poor English input: I eated the purple berries.

Good English output: I ate the purple berries.

Poor English input: Thank you for picking me as your designer. I'd appreciate it.

Good English output: Thank you for choosing me as your designer. I appreciate it.

Poor English input: The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.

Good English output: The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.

Poor English input: I'd be more than happy to work with you in another project.

Prompt examples © Tom B. Brown, et al./ArXiv. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.

We end with a “Poor English” sentence and ...

Prompt

Poor English input: I eated the purple berries.

Good English output: I ate the purple berries.

Poor English input: Thank you for picking me as your designer. I'd appreciate it.

Good English output: Thank you for choosing me as your designer. I appreciate it.

Poor English input: The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.

Good English output: The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.

Poor English input: I'd be more than happy to work with you in another project.

Prompt examples © Tom B. Brown, et al./ArXiv. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.

... and the LLM has learned to fix the error!

Prompt

Poor English input: I eated the purple berries.

Good English output: I ate the purple berries.

Poor English input: Thank you for picking me as your designer. I'd appreciate it.

Good English output: Thank you for choosing me as your designer. I appreciate it.

Poor English input: The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.

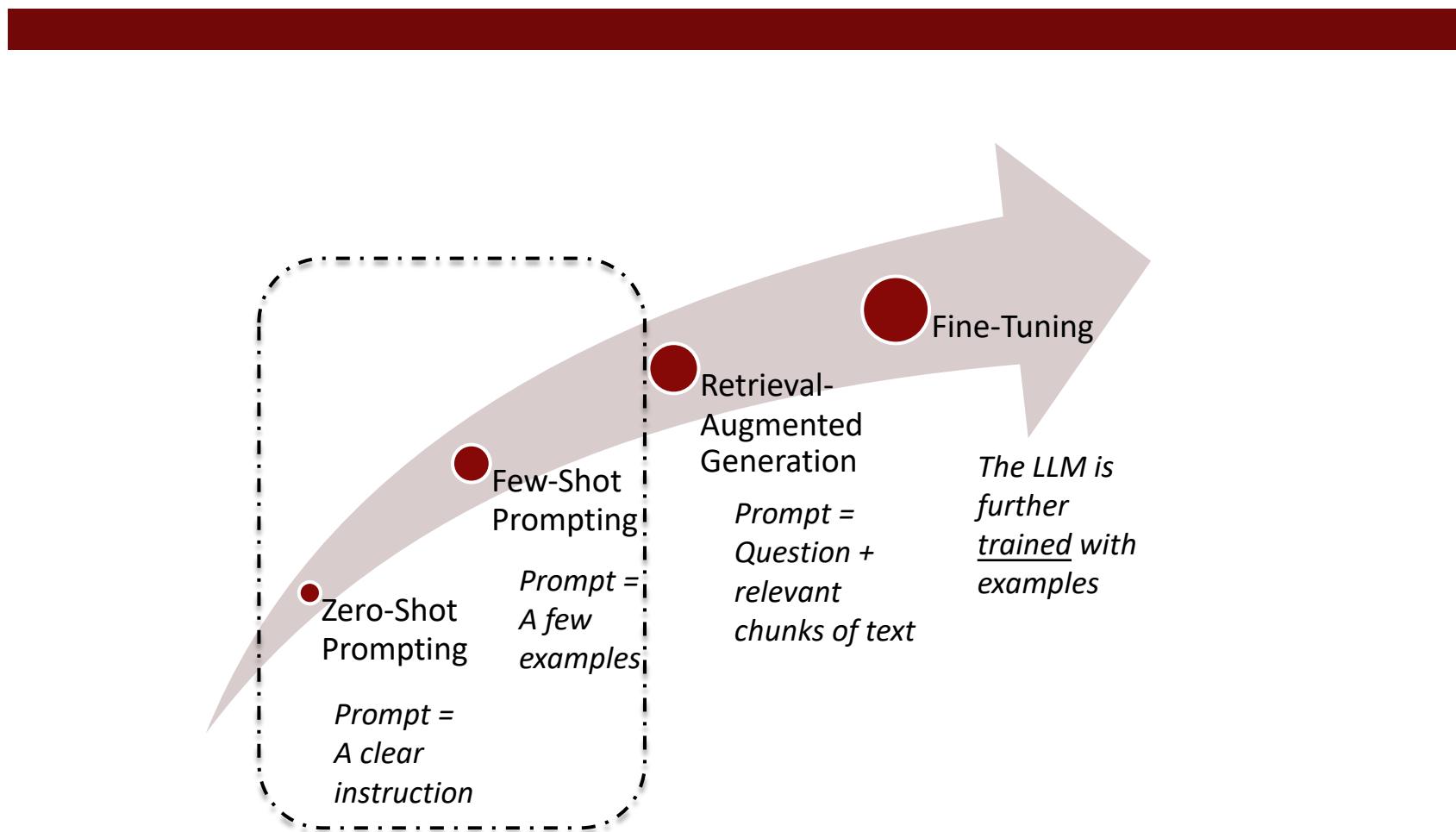
Good English output: The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.

Poor English input: I'd be more than happy to work with you *(in)* another project.

GPT-3

Good English output: I'd be more than happy to work with you *(on)* another project.

The ability of LLMs to learn how to do a task with just instructions/examples in the prompt is called **In-Context Learning**



```
graph LR; A((Zero-Shot Prompting  
Prompt =  
A clear  
instruction)) --> B((Few-Shot Prompting  
Prompt =  
A few examples)); B --> C((Retrieval-Augmented Generation  
Prompt =  
Question +  
relevant  
chunks of text)); C --> D((Fine-Tuning  
The LLM is  
further  
trained with  
examples))
```

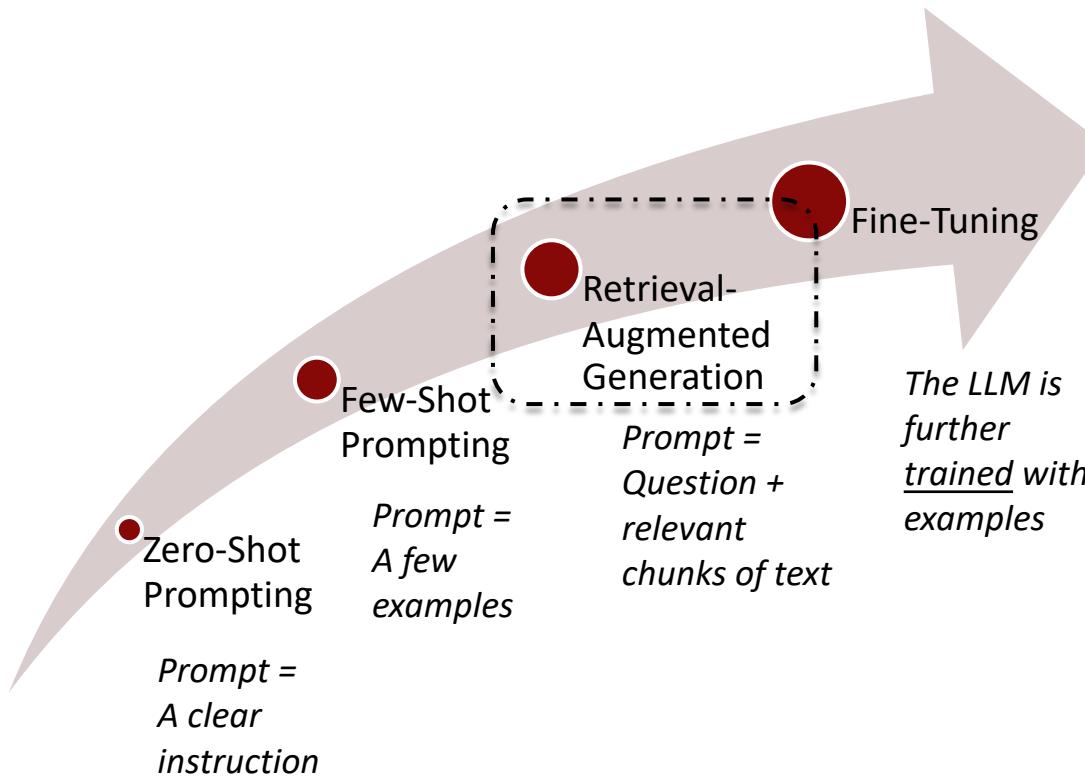
*Prompt =
A clear
instruction*

*Prompt =
A few
examples*

*Prompt =
Question +
relevant
chunks of text*

*The LLM is
further
trained with
examples*

Let's look at Retrieval Augmented Generation* next

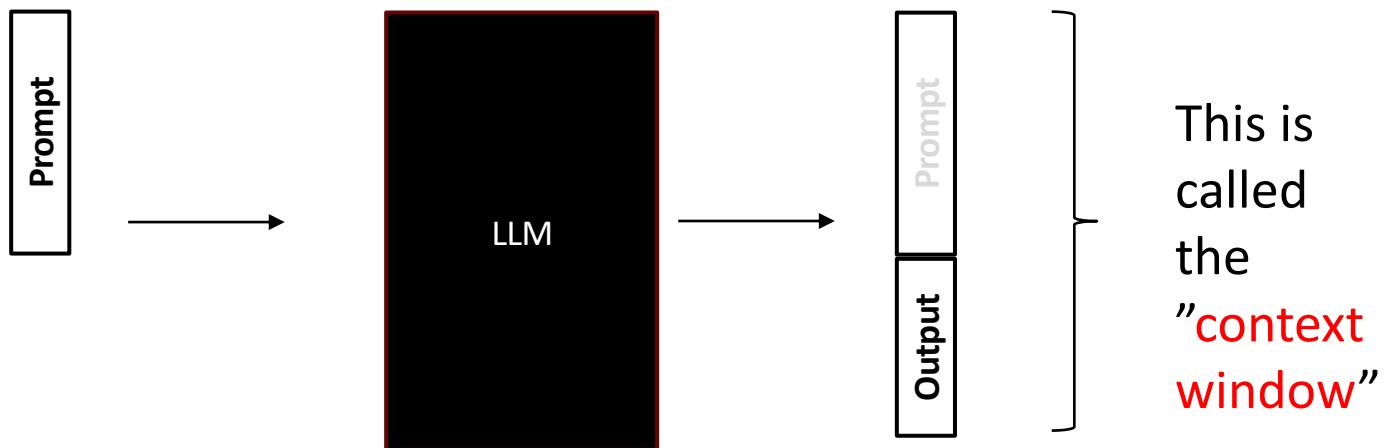


*sometimes referred to as “indexing”

Leveraging proprietary/custom data

- Sometimes we want to leverage proprietary databases and content based on the specific question we want the LLM to answer (e.g., a customer support representative resolving a customer issue).
- Can't we just include the entire dataset in the prompt?

For any LLM, the prompt + output cannot exceed a predefined limit. This is called the **Context Window** (and it varies from LLM to LLM). This limits how much information you can “pack” into your prompt.



Furthermore,
each time you
ask a question,
the entire
conversation so
far is part of
the prompt

What is the square root of 17?

The square root of 17 is approximately 4.123105625617661.

What is it exactly?

The square root of 17 is an irrational number, which means it cannot be expressed exactly as a finite decimal or a fraction. However, it can be represented by the following radical expression:
 $\sqrt{17}$

This is the exact value of the square root of 17 in simplified radical form.

Why did I choose 17 in my example?

As an AI language model, I don't have access to information about why you specifically chose the number 17 for your example. However, 17 is a prime number and a positive integer, so it's a common number to use for mathematical examples and problems. It's also a relatively small number, which makes it easy to work with and calculate the square root by hand or with a calculator.

My favorite number is 17.

That's great to hear! 17 is a prime number and has some interesting mathematical properties. For example, it is a Fermat prime, which means it can be expressed in the form $2^{(2^n)} + 1$, where n is a non-negative integer. It's also the sum of the first four prime numbers ($2 + 3 + 5 + 7 = 17$). Additionally, 17 is a popular number in sports, as it is often worn by athletes in various sports such as football, basketball, and soccer.

A conversation with ChatGPT

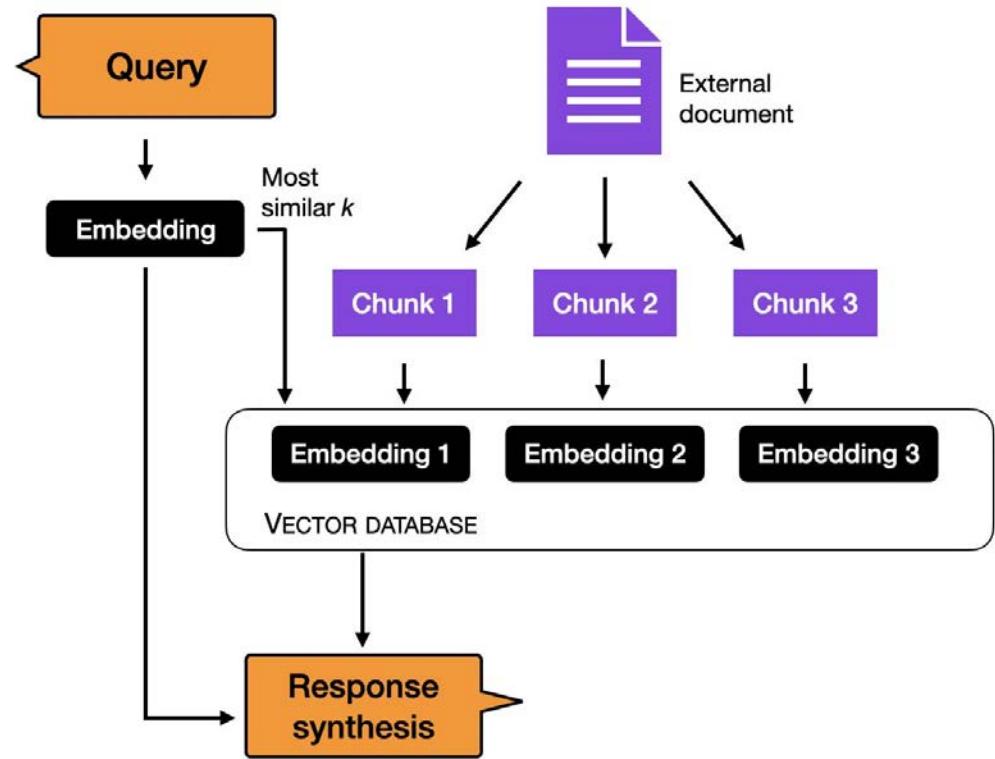
<https://twitter.com/benedwards/status/1644032568772161545?s=20>

Leveraging proprietary/custom data

- Sometimes we want to leverage proprietary databases and content based on the specific question we want the LLM to answer (e.g., a customer support representative resolving a customer issue).
- We can't include the entire dataset in the prompt. We need to first “retrieve” relevant content and send it into the LLM, along with the question.

Retrieval-Augmented Generation

- Sometimes we want to leverage proprietary databases based on the specific question we want the LLM to answer (e.g., a customer support representative resolving a customer issue).
- This can be done by selecting relevant content from an internal database, “packing” it into the prompt and sending it into the LLM.



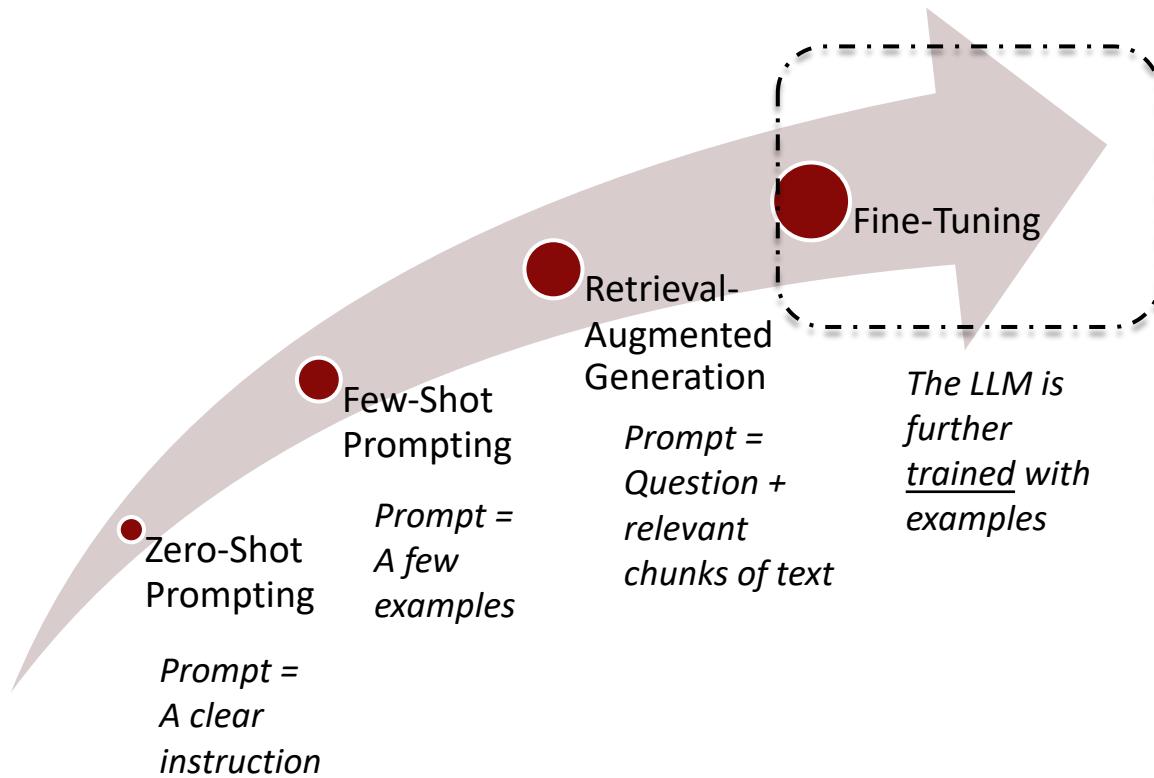
<https://magazine.sebastianraschka.com/p/fine-tuning-large-language-models>

Colab time!

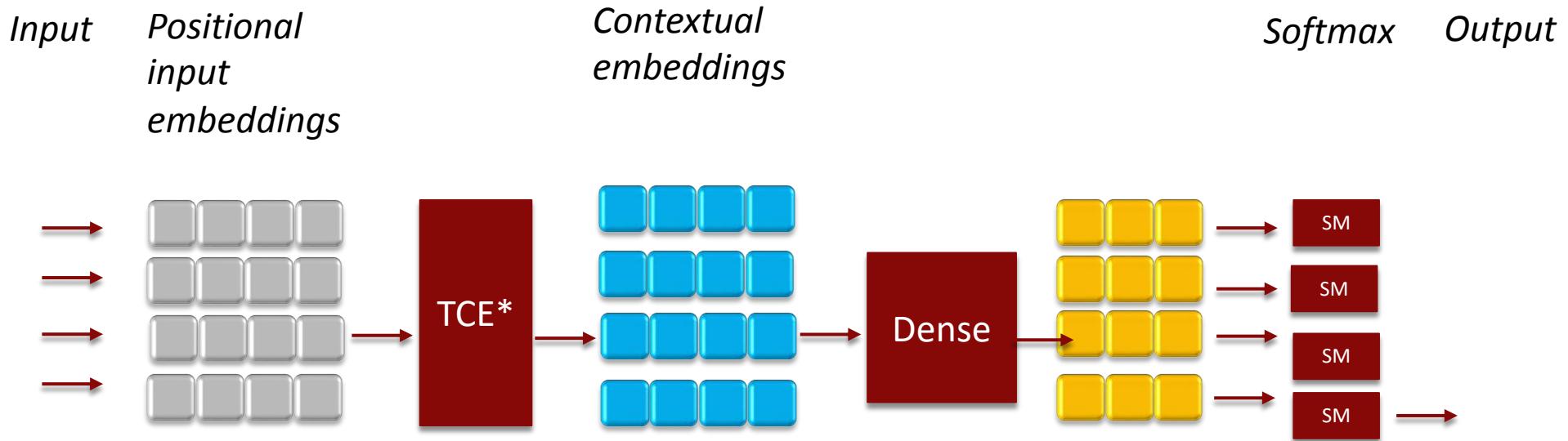


HODL-SP24-Section-A-Lec-10-Retrieval-Augmented-Generation

Let's look at Fine Tuning next

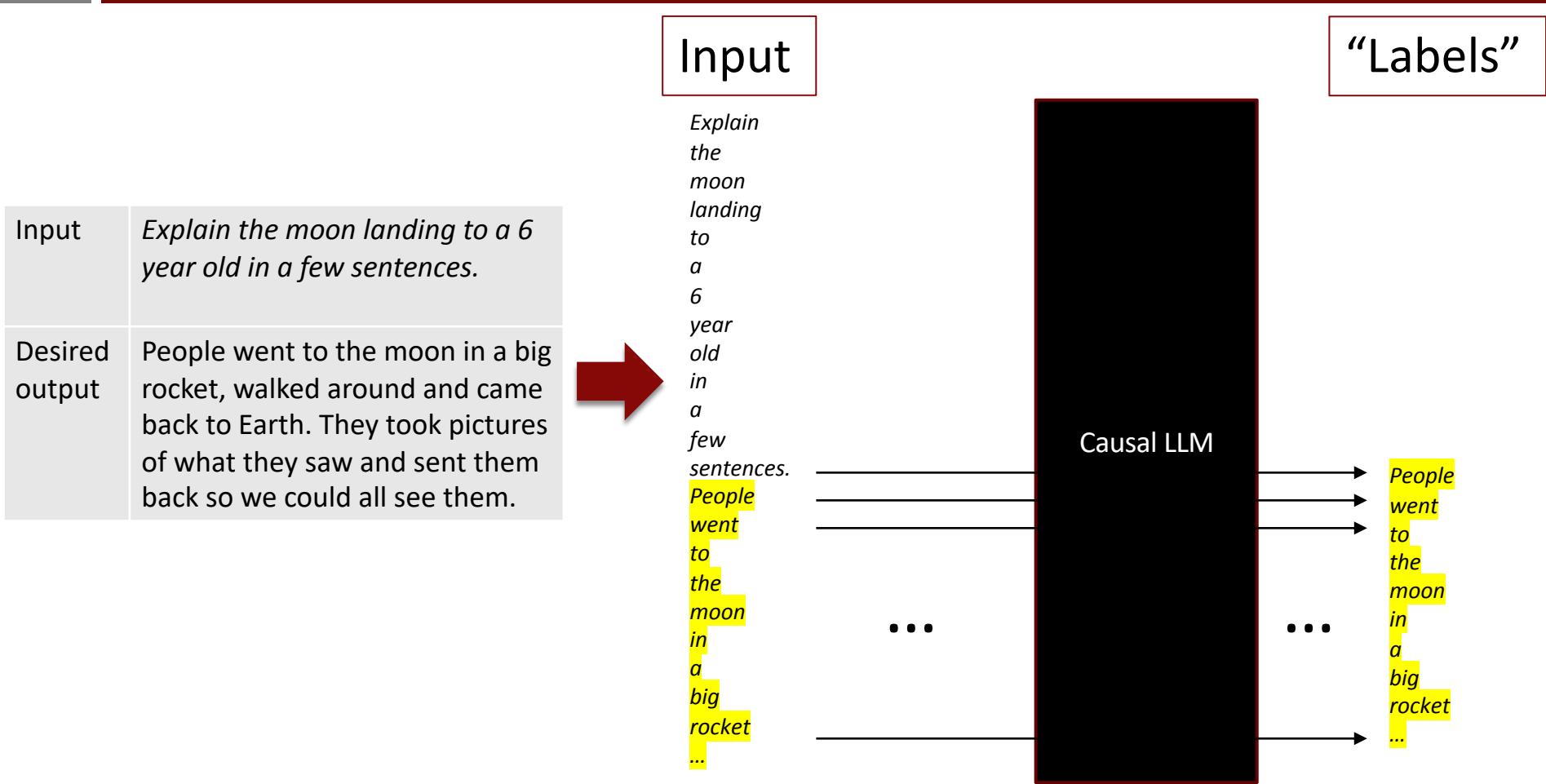


In Fine-Tuning, we take a causal LLM (like GPT) and ...



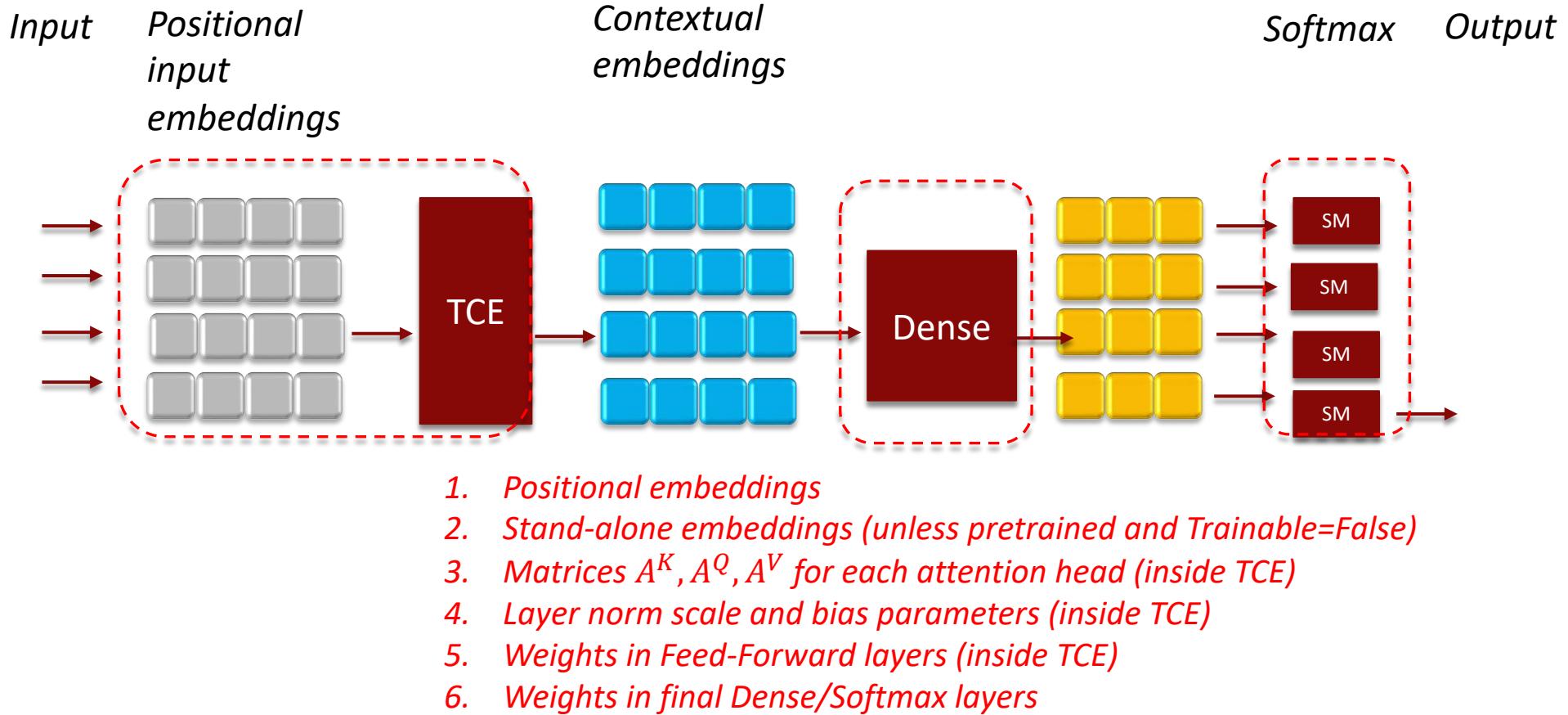
*Transformer **Causal** Encoder stack

... train it further on domain-specific input-output examples ...



* This is essentially the Supervised Fine-tuning step we discussed earlier to transform GPT-3 to GPT-3.5

... and update the weights





Before we discuss the mechanics of fine-tuning, let's look at a quick example of its usefulness

Let's say we ask an LLM to generate synthetic product reviews from product descriptions

Prompt
Write a positive product review using the following information:

rag & bone
NEW YORK

Timeless. Authentic. Iconic. One of the building blocks of a perfect wardrobe. The Fit 2 - our most loved slim-fitting jean. For good reason. Mid-rise, tailored through the hip and thigh with a narrow leg opening. Here in Authentic Stretch: selvedge-inspired denim with an optimal, shape-holding touch of comfort stretch. Featured in raw Japanese unwashed denim.

 Authentic Stretch

 Machine Washable

Size & Fit

Features & Details

98% Cotton 2% Polyurethane

Machine wash inside out with like colors, hang dry

Button fly, mudd tack button, mudd rivets, clean finished hem

Authentic stretch Japanese raw unwashed denim

Made in the United States of America

Product ID: MED23S1223KCRA



LLM Response

Timeless. Authentic. Iconic. One of the building blocks of a perfect wardrobe. The Fit 2 - our most loved slim-fitting jean. For good reason. Mid-rise, tailored through the hip and thigh with a narrow leg opening. Here in Authentic Stretch: selvedge-inspired denim with an optimal, shape-holding touch of comfort stretch. Featured in raw Japanese unwashed denim. Authentic Stretch. Machine Washable. Mid rise slim fit, tailored through hip and thigh

Do you think this reads like a review?

Let's say we ask an LLM to generate synthetic product reviews from product descriptions

Prompt
Write a positive product review using the following information:

rag & bone
NEW YORK

Timeless. Authentic. Iconic. One of the building blocks of a perfect wardrobe. The Fit 2 - our most loved slim-fitting jean. For good reason. Mid-rise, tailored through the hip and thigh with a narrow leg opening. Here in Authentic Stretch: selvedge-inspired denim with an optimal, shape-holding touch of comfort stretch. Featured in raw Japanese unwashed denim.

 Authentic Stretch

 Machine Washable

Size & Fit

Features & Details

98% Cotton 2% Polyurethane

Machine wash inside out with like colors, hang dry

Button fly, mudd tack button, mudd rivets, clean finished hem

Authentic stretch Japanese raw unwashed denim

Made in the United States of America

Product ID: MED23S1223KCRA



LLM Response

Timeless. Authentic. Iconic. One of the building blocks of a perfect wardrobe. The Fit 2 - our most loved slim-fitting jean. For good reason. Mid-rise, tailored through the hip and thigh with a narrow leg opening. Here in Authentic Stretch: selvedge-inspired denim with an optimal, shape-holding touch of comfort stretch. Featured in raw Japanese unwashed denim. Authentic Stretch. Machine Washable. Mid rise slim fit, tailored through hip and thigh

No, this reads like Marketing copy.

Another example

Prompt
Write a positive product review using the following information:

URBAN OUTFITTERS

Details

Product Sku: 81419525; Color Code: 083

Crew neck jersey tank top by Champion. Heritage style tee with a cotton fabrication and crew neck. Urban Outfitters exclusive.

Features

- Sleeveless tee from Champion
- Boxy and cropped silhouette
- Crew neckline

Content + Care

- 100% Cotton
- Machine wash
- Imported

Size + Fit

- Model is 5'9.5" and wearing size Large
- Measurements taken from size Large
- Chest: 25.75"
- Length: 22.25"

Champion

Since 1919, Champion has been offering its full line of innovative athletic wear, from iconic tees and sweatshirts to hats and gear.



LLM Response

Champion's Crew Neck Jersey Tank Top is a great addition to any wardrobe. The boxy and cropped silhouette is flattering on all body types, and the cotton fabrication is soft and comfortable. The crew neckline is classic and timeless, and the sleeveless design makes it perfect for warmer weather. The Heritage style tee is an exclusive from Urban Outfitters, and the 100% cotton fabrication and machine washability make it easy to care for.

This too reads like Marketing copy.



Let's fine-tune the LLM with
[Instruction,
Product description,
Product review] examples!

A dataset for instruction tuning (example)

```
{  
  "instruction": "Write a positive review for the following product",  
  "input": "The Workwear Pocket Long Sleeve T-shirt. Wear it as a layer, wear it by itself, wear it with pride. It's as rugged and dependable as your favorite hammer offering 6.75-ounce, 100% cotton jersey knit, a generous left-chest pocket, side-seamed construction to minimize twisting and crew neck comfort. Made of 6.75-ounce, 100% cotton jersey our long sleeve pocket t-shirt makes for perfect workwear layered or worn alone. It features a crew neck with a generous left chest pocket and side-seamed construction to minimize twisting. Ash fiber content is 99%/1% polyester. Heather gray fiber content is 90% cotton/10% polyester. 100% Cotton Imported Machine Wash Long-sleeve t-shirt in midweight jersey featuring spade patch pocket at chest with logo tag Ribbed cuffs Original fit Package Dimensions:\n          \n12.4 x 7.6 x 1.8 inches Shipping Weight:\n          \n1 pounds (View shipping rates and policies)",  
  "output": "The Best! Carhartt t-shirts are my husband's favorite. They fit well, the fabric is very substantial, they wear well, and he especially likes the pocket. He wears these alone, with a light t-shirt under, or under a long sleeve button down Carhartt work shirt."  
},  
  
{  
  "instruction": "Write a neutral review for the following product",  
  "input": "These overalls are made of 12-ounce, firm-hand 100 percent ring-spun cotton duck and have adjustable front-elastic suspenders, a multi-compartment bib pocket, a zipper fly, two large lower-front pockets a left-leg hammer loop, double-front knee construction with cleanout bottoms that accommodate knee pads, two reinforced back pockets, triple-stitched main seams and metal rivets at vital stress points. Care-Do not bleach Our duck bib overall is made of heavy 12-ounce, cotton duck. It features a variety of tool and utility pockets as well as a hammer loop. Double knees can accommodate knee pads and feature cleanout bottoms. Back pockets are reinforced for durability. 100% Cotton Imported Machine wash warm with like colors, Do not bleach, Tumble dry medium remove promptly Bib overall in 12-ounce heavyweight cotton duck featuring hammer loop and reinforced paneling at knees Side utility pockets and bib pocket with logo patch double knees Kindly refer last Image for sizing details Multiple tool pockets Package Dimensions:\n          \n15.6 x 6.3 x 3.5 inches Shipping Weight:\n          \n2.95 pounds (View shipping rates and policies)",  
  "output": "... a little leary about a new pair being as good because. I have a 15 year old pair of these that got a hole burned through them when using a cut off saw and was a little leary about a new pair being as good because... well, I am in my 50's and have seen the vast majority of consumer/professional goods turn to crap. So far so good."  
},  
  
{  
  "instruction": "Write a negative review for the following product",  
  "input": "The iconic, timeless chuck taylor all star sneaker. Perfect in its simplicity since 1917. Lightweight, breathable canvas construction. Vulcanized rubber sole delivers durable traction. An ortholite insole cushions each and every step. Versatile, fashionable, and they keep looking better every time you wear them. Ortholite is a trademark of O2 partners, llc. 100% Textile Imported Rubber sole Shaft measures approximately low-top from arch Lace-up, low-top sneaker OrthoLite insole for cushioning Medial eyelets for airflow Canvas upper Product Dimensions:\n          \n5 x 5 x 0.7 inches Shipping Weight:\n          \n12 ounces (View shipping rates and policies)",  
  "output": "I love the classic look of Chuck Taylors I love the classic look of Chuck Taylors, but wow, these are legitimately the most uncomfortable shoes I've ever worn. I wore them recently to my work where I'm on my feet for 10 hours and am moving around a lot and lifting stuff, climbing up and down ladders, and by the end of the night I was limping and my feet were throbbing. I have another pair of Chuck Taylors, mid tops, and those have some padding in the sole and I can wear them all night and feel fine. I guess I'm going to have to get some inserts for these if I want to wear them longer than an hour."  
},
```

The results of (Supervised) Finetuning

rag & bone
NEW YORK

Timeless. Authentic. Iconic. One of the building blocks of a perfect wardrobe. The Fit 2 - our most loved slim-fitting jean. For good reason. Mid-rise, tailored through the hip and thigh with a narrow leg opening. Here in Authentic Stretch: selvedge-inspired denim with an optimal, shape-holding touch of comfort stretch. Featured in raw Japanese unwashed denim.

 Authentic Stretch

 Machine Washable

Size & Fit

Features & Details

98% Cotton 2% Polyurethane

Machine wash inside out with like colors, hang dry

Button fly, mudd tack button, mudd rivets, clean finished hem

Authentic stretch Japanese raw unwashed denim

Made in the United States of America

Product ID: MED23S1223KCRA

LLM Responses

Write a review:

These are the best jeans I've ever owned. I'm 5'10" and 160 lbs. I bought a size 32 and they fit perfectly. I've been wearing them for a few months now and they still look brand new. I've washed them a few times and they've held up great. I've never owned a pair of jeans that fit this well and look this good after a few months of wearing them. I'll definitely be buying more of these in the future.

Write a horrible review:

These are the worst jeans I've ever owned. I'm 5'10" and 160 lbs. I ordered a 32 and they fit like a 30. They're too tight in the thighs and too loose in the waist. I'm going to return them and try a 30, but I'm not optimistic. I've been wearing Levi's 501s for 30 years and these are not even close to the quality of those jeans. I'm going to stick with Levi

These read like real reviews!

The results of (Supervised) Finetuning

URBAN OUTFITTERS

Details

Product Sku: 81419525; Color Code: 083

Crew neck jersey tank top by Champion. Heritage style tee with a cotton fabrication and crew neck. Urban Outfitters exclusive.

Features

- Sleeveless tee from Champion
- Boxy and cropped silhouette
- Crew neckline

Content + Care

- 100% Cotton
- Machine wash
- Imported

Size + Fit

- Model is 5'9.5" and wearing size Large
- Measurements taken from size Large
- Chest: 25.75"
- Length: 22.25"

Champion

Since 1919, Champion has been offering its full line of innovative athletic wear, from iconic tees and sweatshirts to hats and gear.

Champion's Crew Neck Jersey Tank Top is a great addition to any wardrobe. The boxy and cropped silhouette is flattering on all body types, and the cotton fabrication is soft and comfortable. The crew neckline is classic and timeless, and the sleeveless design makes it perfect for warmer weather. The Heritage style tee is an exclusive from Urban Outfitters, and the 100% cotton fabrication and machine washability make it easy to care for. The boxy and cropped silhouette is flattering on

I love this tank top! It's so soft and comfy, and the fit is perfect. I'm 5'7" and 120 lbs, and I got a small. It's not too tight, but it's not too loose either. It's a great tank top to wear to the gym or just around the house. I would definitely recommend this tank top to anyone who is looking for a comfy tank top to wear around the house or to the gym.

These read like real reviews!



For small causal LLMs (like GPT-2), fine-tuning isn't difficult but for larger LLMs, we will face computational challenges.

We will explain the nature of these challenges and describe an alternative approach that addresses these challenges.

The Llama 2 family of open LLMs are widely used for fine-tuning

Large language model

Llama 2: open source, free for research and commercial use

We're unlocking the power of these large language models. Our latest version of Llama – Llama 2 – is now accessible to individuals, creators, researchers, and businesses so they can experiment, innovate, and scale their ideas responsibly.

[Download the model](#)



<https://llama.meta.com/llama2>

Let's first understand how "hard" it is to build the biggest model in the family: Llama-2-70b

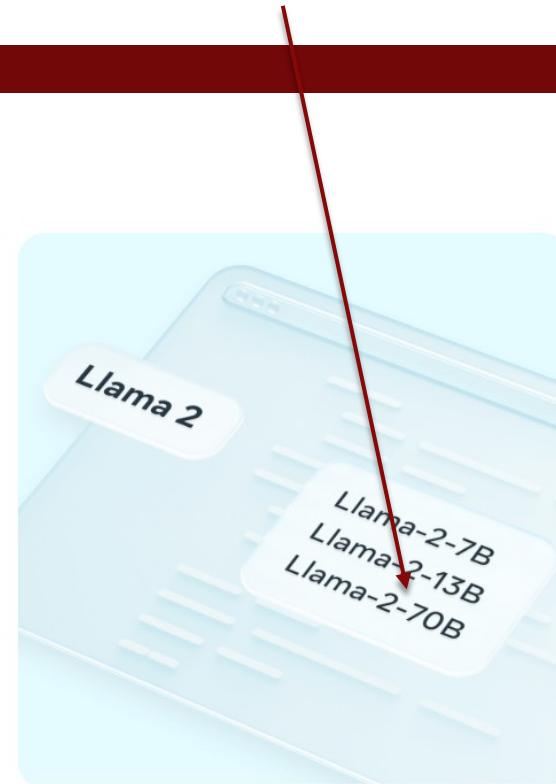
Large language model

Llama 2: open source, free for research and commercial use

We're unlocking the power of these large language models. Our latest version of Llama – Llama 2 – is now accessible to individuals, creators, researchers, and businesses so they can experiment, innovate, and scale their ideas responsibly.

[Download the model](#)

<https://llama.meta.com/llama2>



How hard is it to train Llama-2-70b?

- The model is gigantic
 - 70 billion parameters x 2 bytes per parameter x ~3-6 = **420-840GB**
 - 2 bytes/parameter assuming we use “fp16” (i.e., 16-bit numbers)
 - Overall, 3-6x multiplier for each parameter since we need to store gradient, optimizer states etc. in addition to the weights themselves, and some require higher precision (more on this later)
- An A100 (or H100) GPU has 80GB of RAM and so **we need between 6 and 11 GPUs to accommodate 420-840GB**

How hard is it to train Llama-2-70b?

- Llama-2-70b was trained on 2 trillion (2×10^{12}) tokens
- An A100 GPU can (optimistically) process just about 400 tokens/GPU per second
- 11 GPUs will take $2 \times 10^{12}/(11 \times 400)$ seconds which is about 5,261 days.
- Let's say we want to do it in about a month.
 - With 2048 GPUs, we would need about 28 days
 - A simple cost estimate (at \$2.5/ GPU-hr) for this training run is \$4M
 - We would expect the actual cost to be a lot higher since it takes multiple runs to get things right

To fine-tune Llama-2-70b with **fewer resources**, we need to do two things

- Reduce the size of the dataset
- Reduce the memory required so that we can process the data using fewer GPUs (ideally, just one GPU)

To fine-tune Llama-2-70b with fewer resources, we need to do two things



- Reduce the size of the dataset: We are in luck here! Finetuning datasets can be much smaller than the original corpus used to train the LLM in the first place
 - The famous Alpaca fine-tuning dataset has 50k instruction-answer pairs at 4096 tokens each, so $\sim 200M$ tokens in total, which is <<< 2T tokens used to train Llama-2-70b
 - Assuming we can process 400 tokens/GPU per second, 7 GPUs will take $2 \times 108/(7 \times 400)$ seconds which is only about 20 hours (as opposed to 28 days for training Llama-2-70b)!
- Reduce the memory required so that we can process the data using fewer GPUs (ideally, just one GPU)

Next, let's discuss how to reduce the memory requirements

- Reduce the size of the dataset: We are in luck here! Finetuning datasets can be much smaller than the original corpus used to train the LLM in the first place
 - The famous Alpaca fine-tuning dataset has 50k instruction-answer pairs at 4096 tokens each, so $\sim 200M$ tokens in total, which is <<< 2T tokens used to train Llama-2-70b
 - Assuming we can process 400 tokens/GPU per second, 7 GPUs will take $2 \times 108/(7 \times 400)$ seconds which is only about 20 hours (as opposed to 28 days for training Llama-2-70b)!
- Reduce the memory required so that we can process the data using fewer GPUs (ideally, just one GPU)

What consumes memory?

	Naive Memory Usage	Optimized
Model parameters	#Params x 2Bytes = 140GB	
Gradient computations	Same as above = 140 GB	
Optimizer state	1-4x the memory needed for parameters = 140-560GB	
Total	420-840GB	

- 
- It turns out that state-of-the-art optimizers (like Adam) need to store information related to past gradients
 - In fact, this requires approximately 1-4x the amount of memory in addition to the parameters and the gradient

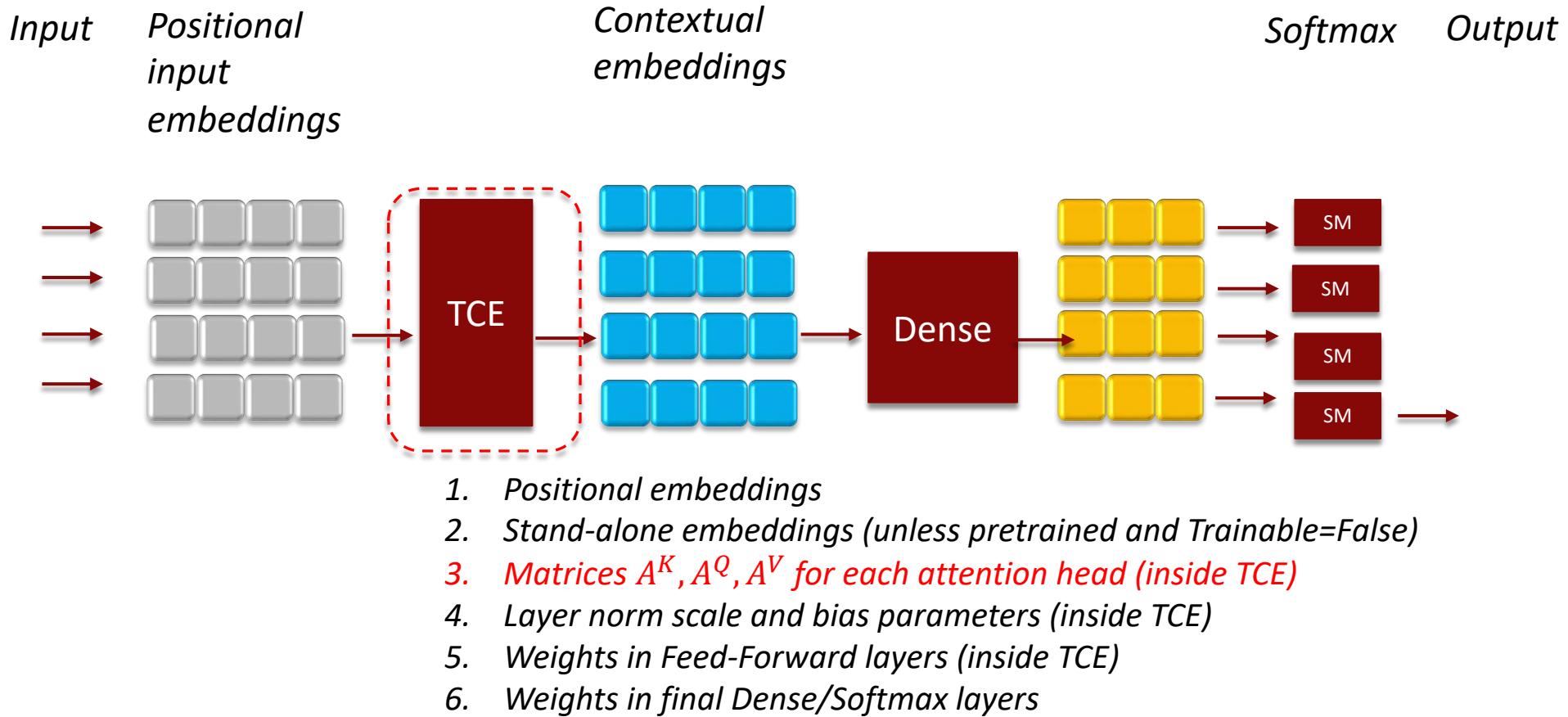
What consumes memory?

	Naive Memory Usage	Optimized
Model parameters	#Params x 2Bytes = 140GB	140GB*
Gradient computations	Same as above = 140 GB	~ zero by computing 'just in time'**
Optimizer state	1-4x the memory needed for parameters = 140-560GB	We will show how to reduce this to ~ zero!
Total	420-840GB	140GB

*This can be reduced with 'quantization' but this often can result in performance degradation

** This requires an old trick called 'gradient checkpointing' that is beyond the scope of our discussion here.

We will fine-tune only the matrices inside the causal self-attention blocks, and will keep everything else frozen



Consider the weight matrix A^K

- In Llama-2-70b, there are 64 self-attention heads in every self-attention layer.
- The A^K matrix for each head is 8192×128 but since there are 64 heads, we can imagine combining all 64 matrices into one big $8192 \times (128 * 64) = 8192 \times 8192$ matrix
- Thus, in each self-attention layer, we have ~ 64 M parameters to store just for A^K .

An update to A^K can be thought of as the sum of the original A^K and the change ΔA^K

1.1	0.2	-0.7	2.3	-0.3
-0.4	-0.9	2.0	3.7	2.1
0.4	-1.2	0.3	2.8	1.3
2.3	-0.3	-0.9	2.0	0.1
0.2	-0.7	-0.5	-1.2	0.3

0.01	0.02	0.04	-.01	-.03
-.04	-.09	-.04	-.09	0.02
0.01	-.02	0.04	-.08	.07
-.05	-.01	-.09	.02	0.01
0.07	-.01	-.05	-.12	.03

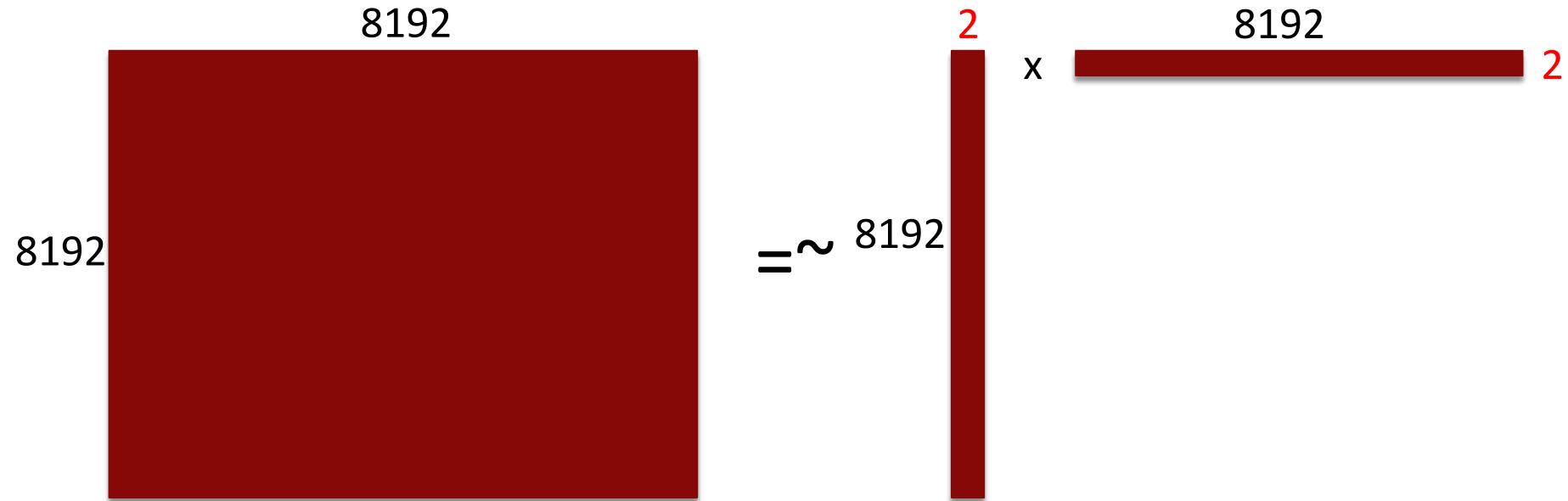
$$A^K + \Delta A^K$$

- In general, the change matrix ΔA^K will be as big as A^K
- **Can we make it smaller?**

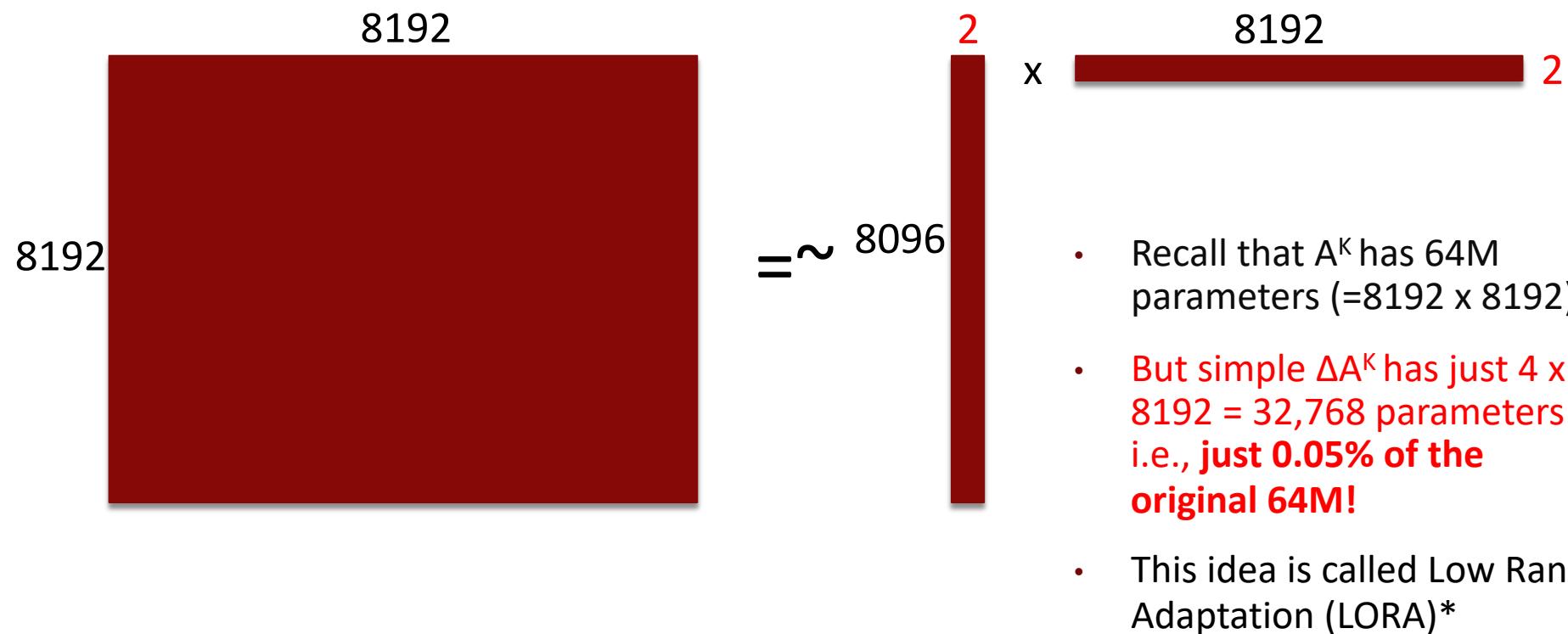


A finetune will likely make very small changes to the original model weights. This suggests that we can “force” the change matrix ΔA^K to be “simple” and it will still get the job done.

We can force ΔA^K to be “simple” by forcing to be “low rank”



The number of parameters in “simple” ΔA^K



LoRA Optimization

- Freeze all base model parameters
- Initialize $\Delta A^K \Delta A^Q \Delta A^V$ to zero in each self attention layer
- Update $\Delta A^K \Delta A^Q \Delta A^V$ (via SGD as usual) by updating the two “skinny” matrices for each

What does memory look like now?

	Naive Memory Usage	Optimized
Model Parameters	#Params x 2Bytes = 140GB	The same*
Gradient Computations	~#Neurons ~ #Params ~140GB	Roughly zero by computing these 'just in time'**
Optimizer State	#Params x 1-4Bytes x 2 = 140-560GB	Using LoRA, we only need to optimize the two "skinny" matrices so # of parameters ~0
Total	420-840GB	140GB

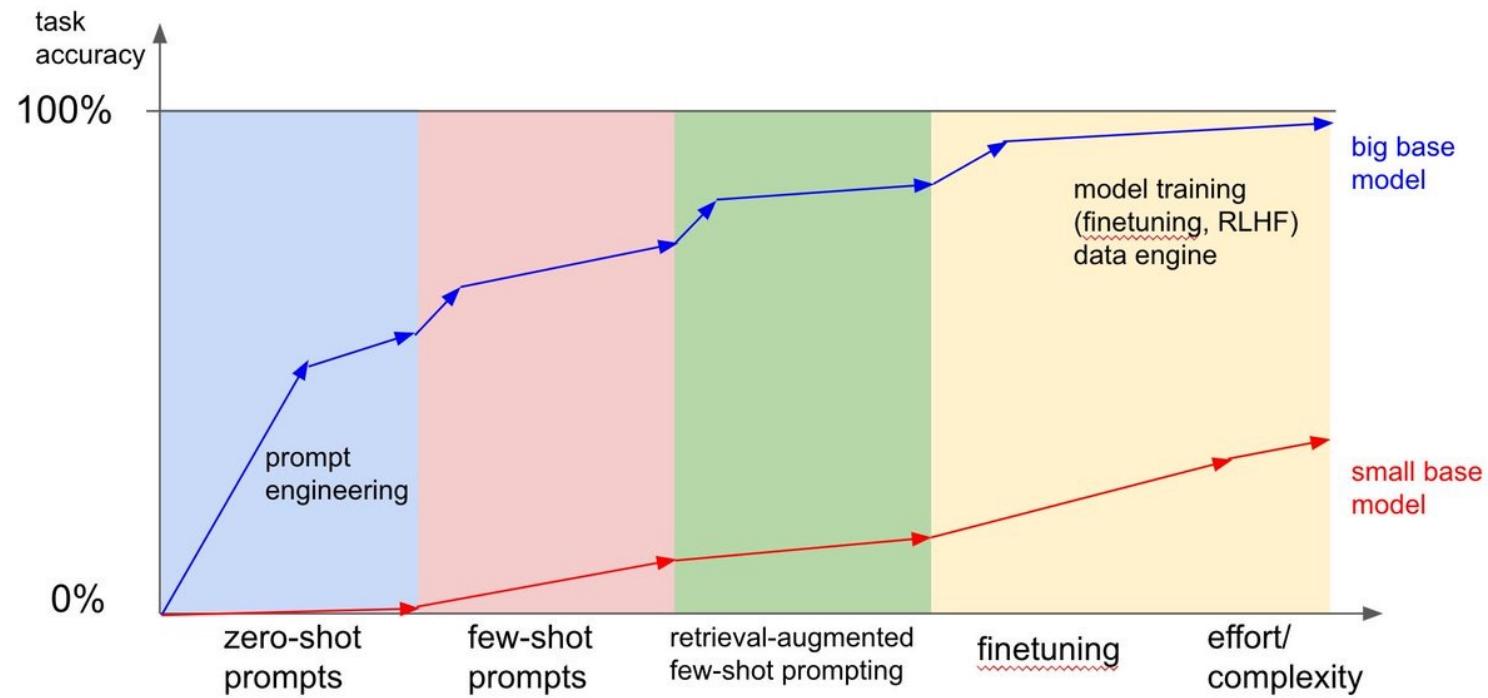
Now, Llama-2-70b can be comfortably be finetuned on 2 GPUs and Llama-2-7b and Llama-2-13b can comfortably be finetuned on a single GPU.

LoRA Finetuning Colab



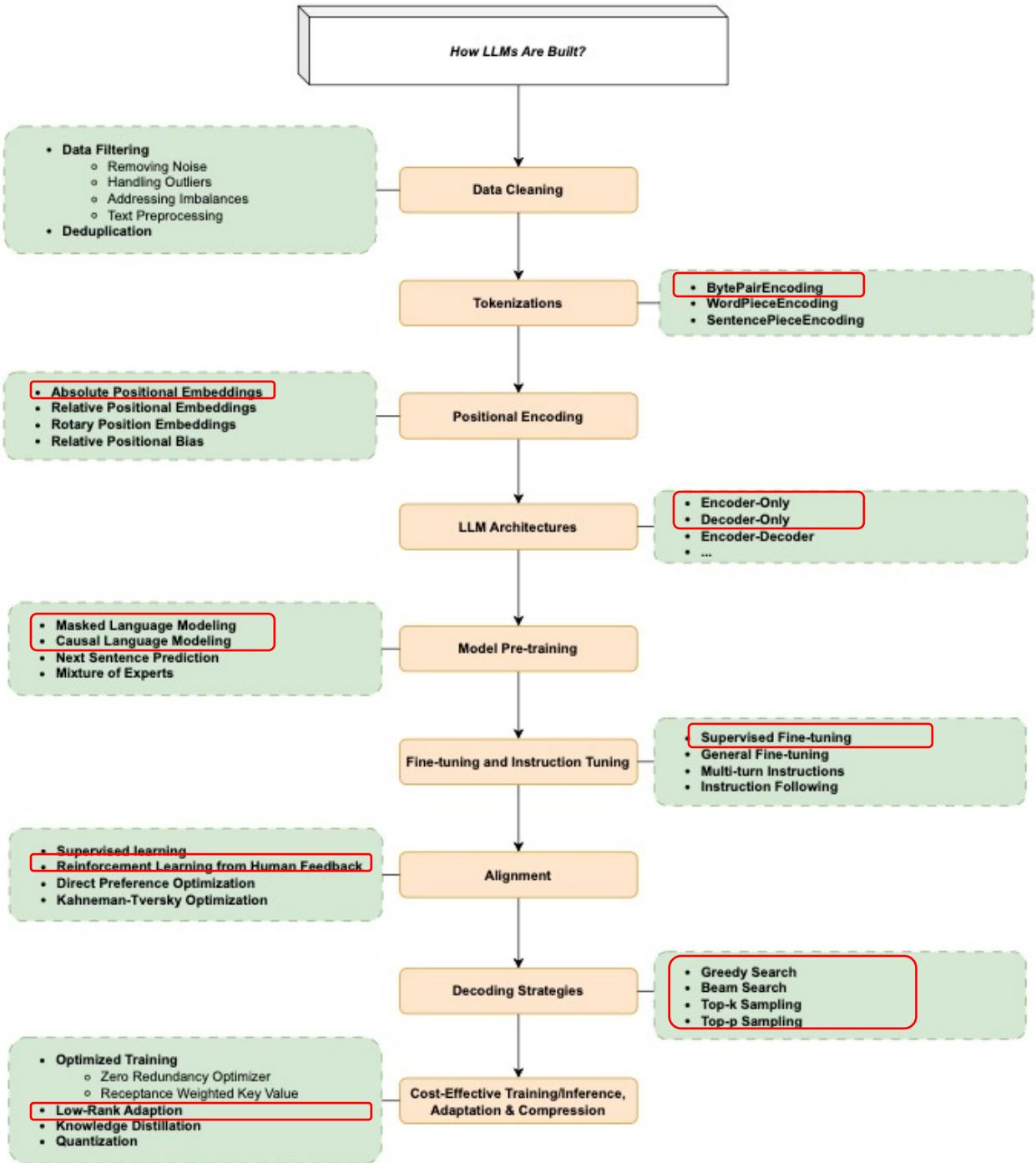
Stay tuned for video and colab.

The effort-benefit curve for adaptation strategies depends on the size of the base LLM



Effort-benefit curve figure © Andrej Karpathy on Twitter. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use>.

We have
covered all
these topics!



How LLMs are built figure by Shervin Minaee,
Tomas Mikolov, Narjes Nikzad, Meysam
Chenaghlu, Richard Socher, Xavier
Amatriain, and Jianfeng Gao. License: CC BY.

MIT OpenCourseWare
<https://ocw.mit.edu>

15.773 Hands-on Deep Learning

Spring 2024

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.