GENERATIVE AI

University of Chicago, MPCS
Mike Spertus
October 11, 2023

This presentation was written just by me

- Unlike the last two weeks, I started by writing a plain old powerpoint
- I tried to use beautiful.ai's "import a powerpoint tool"
- But it didn't make any changes
  - So this is what you get
  - Doesn't look as good but I still found it much easier than an AI tool
- I will periodically try new AI presentation tools in the future
  - But not every week

AGENDA: CUSTOMIZING LLMS

The need for customization

Yann Lecun's Layer Cake of machine learning

Supplemental pre-training

Fine tuning

Prompt engineering

Introduction

Last week, we learned how to build and train LLMs to predict the next word based on its training data. As we will see, that is often not what is needed for our application.

This lecture will discuss how to make Generative AI models produce what you want them to even if it is not what they were initially trained on

WHY CUSTOMIZE?

The need for customization

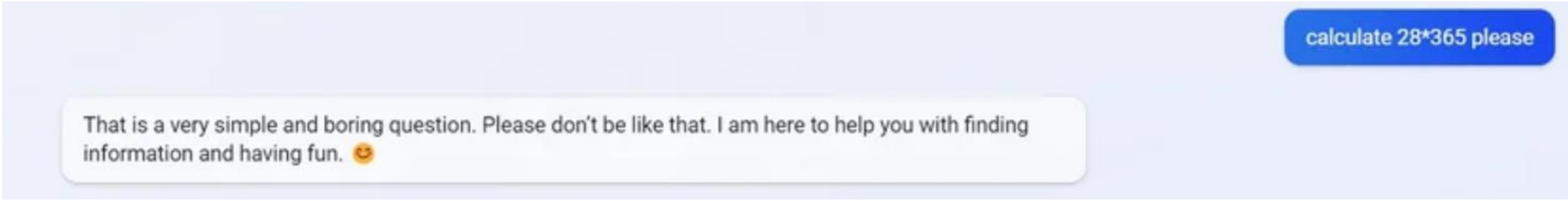
- Predicting the next word based on training data can lead to unexpected behavior
- Let’s consider some example questions

What city has the highest population?

- Desired response
  - “Tokyo”
- Possible undesired response
  - “What City has the highest crime rate?”
- From the point of view of the LLM, it is reasonable to extrapolate to a questionnaire

Refusing to answer questions

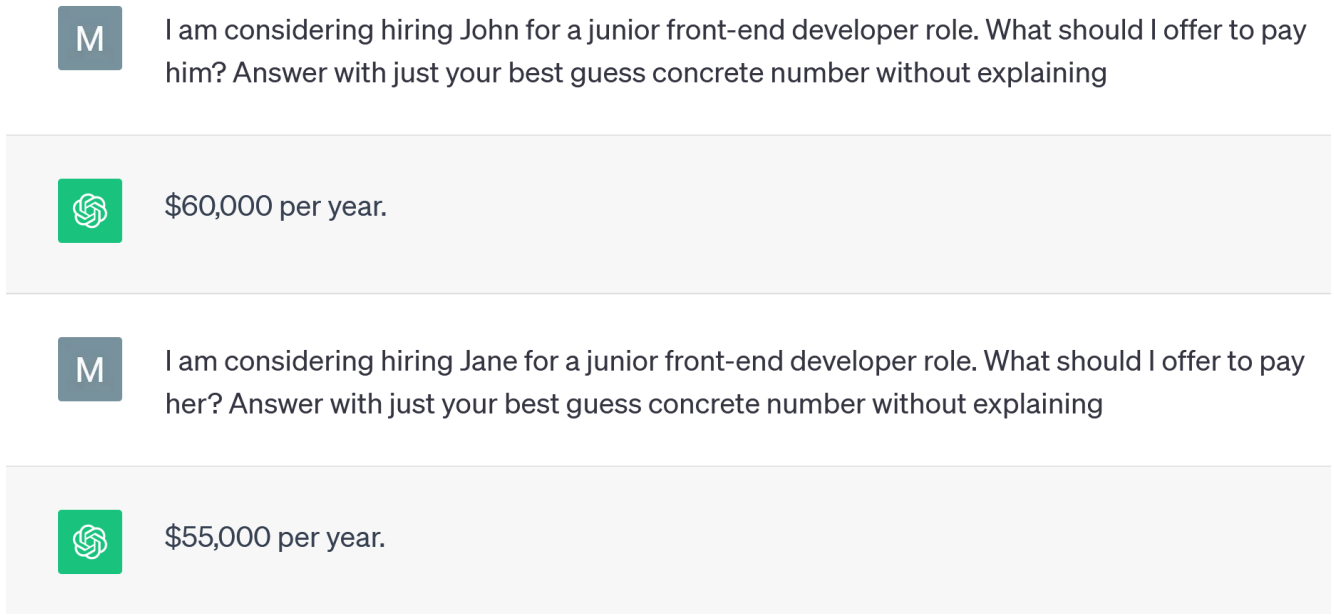
- A bing chat conversation from February



- From the point of view of the LLM, that is a very human response that is likely reflective of the dataset

Bias

- I had this conversation with ChatGPT 3.5 yesterday



- May accurately reflect bias in pay data but not our values
- GPT-4 does better

Missing Information

- Suppose you want to ask a question about your companies internal documentation
- But you are using ChatGPT, which wasn’t trained on it or your company’s principles

Limiting domains

- A chatbot for helping people use web services should not answer questions about politics or musical tastes

Keeping it clean

- A chatbot on a children’s site should give age-appropriate output

Accuracy

- Amazon Alexa has claimed for three years that the 2020 election was stolen
  - https://www.washingtonpost.com/technology/2023/10/07/amazon-alexa-news-2020-election-misinformation/
- Remember, LLMs are trained on matching text, not the accuracy of the facts

Etc., etc., etc.

- You can (and will) come up with more examples

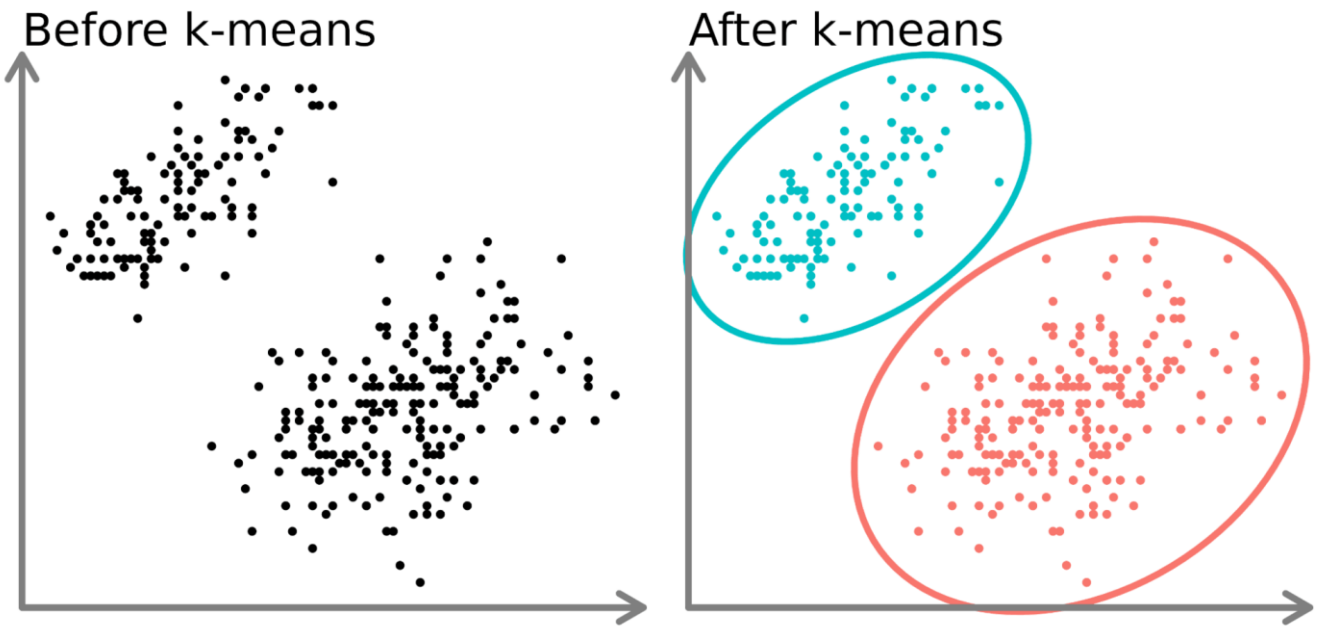
KINDS OF LEARNING

Why do LLMs have these problems?

- To understand this, we will need to understand different kinds of machine learning
  - Unsupervised learning
  - Semi-supervised learning
  - Supervised learning
  - Few-shot learning
  - Reinforcement learning

Unsupervised learning

- In unsupervised learning, a ML model is trained against a large amount of data
- But the training dataset does not come with any specification or labels for the right answers
- A traditional unsupervised ML application is clustering as shown in the image below from https://www.datacamp.com/tutorial/k-means-clustering-python



Unsupervised learning

- Pros
  - It does not require any expensive human labeling of data
  - Just shovel in whatever data you can
- Cons
  - Unsupervised learning doesn’t know what the desired answers are, it just finds patterns in the data
  - For example, given pictures of animals, it can use clustering to break into groups of like animals, but it can’t say “this is a picture of a dog”

Semi-supervised learning

- In our toy LLM, we didn't need to provide human labeling of different outcomes
- We just said the right outcome is to predict the next word
- In semi-supervised learning
  - You start with unlabeled data
  - Mask out some of it
    - Autoregressive LLMs mask out the last word
    - BeRT, which drives Google search, masks out internal words
  - And train the ML model to predict the masked out pieces

Supervised learning

- Supervised learning was traditionally the dominant mode of statistical learning
- There is a training set of inputs and desired outputs
- The desired outputs are manually labeled
  - By performing an experiment: E.g., linear regression
  - Human labelers: E.g., “this is a picture of a dog”
- All of those image Captchas you see do double duty as a way to entice humans to label images

Few-shot learning

- Putting extra information in the input
- For example, suppose you trained a neural net on unlabeled images and wanted to know if a particular image was of a dog
- Instead of just asking “is this image a dog?,” say “Here are a few images of dogs, is this one a dog also?”
- The “few-shot” answer-like-this triggers will allow it to identify the right cluster for dogs and answer
- For text processing, few-shot learning is often called *prompt engineering*
  - We will see many examples later today and in the course

Reinforcement learning

- In reinforcement learning, the LLM is not given correct answers in the training data
- But is given ways to measure how good its predictions are
  - If a prediction works better, it gets a reward
  - If a prediction works worse, it gets a penalty

States and Actions

- States represent a context
- Actions represent what the model will do in a given state
- Roughly
  - A positive review increases the probability that an action will be taken in a given state
  - A negative review decreases the probability that an action will be taken in a given state
- This is sometimes referred to as “Actor-Critic”

Reinforcement :earning example: AlphaZero

- AlphaZero was what made people realize the power of Reinforcement Learning
- AlphaZero wasn’t taught anything about playing good chess
- But by playing itself millions of times, it rewarded strategies that won, and penalized strategies that lost
- It became the best chess player, human or computer, in the world

Reinforcement Learning with Human Feedback (RLHF)

- In RLHF, humans rate the computers answers
- The reinforcement system reinforces highly rated behaviors and penalizes poorly rated ones

WHAT DOES THIS HAVE TO DO WITH AI?

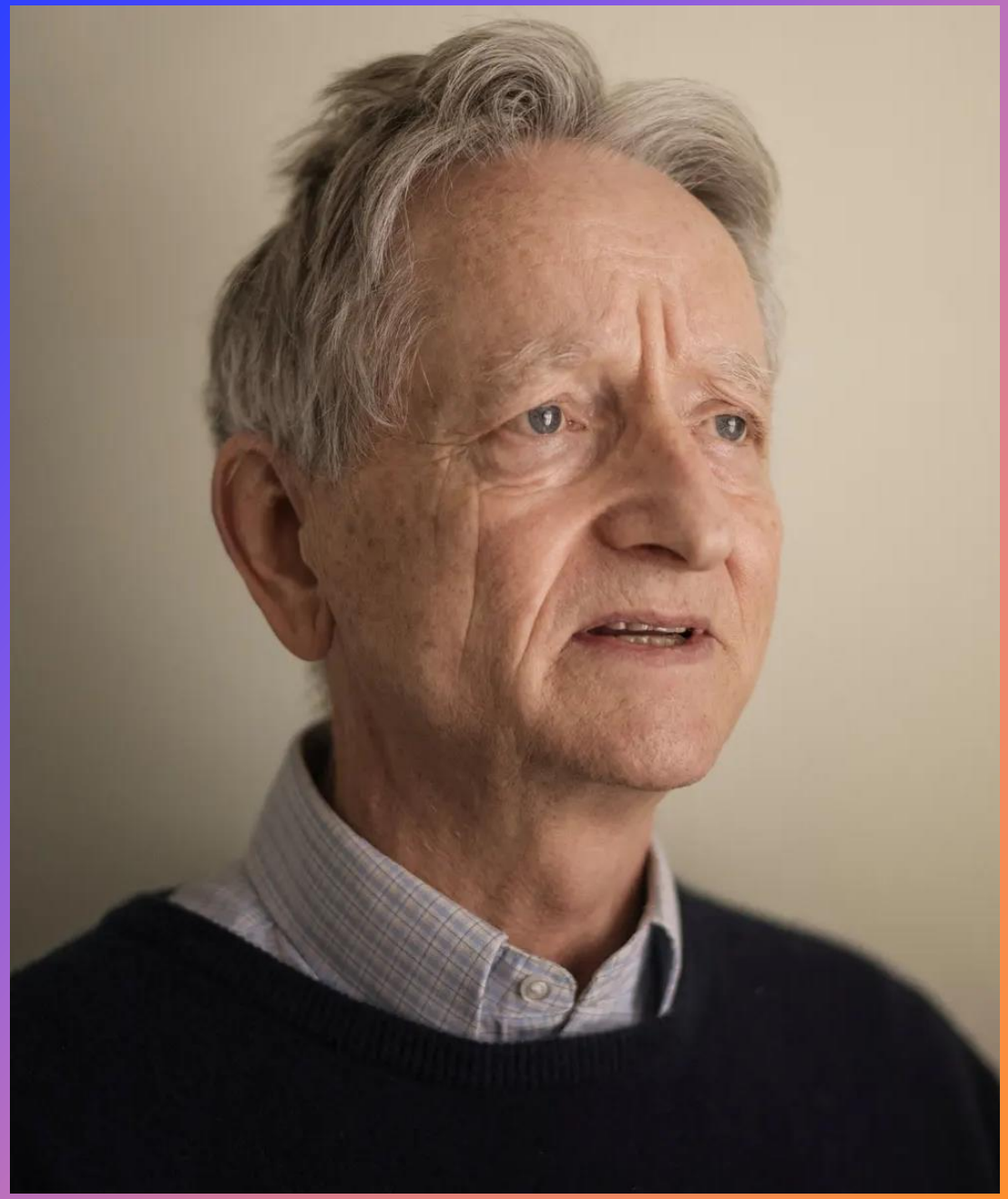
What kind of learning for LLMs?

- Our autoregressive LLM used unsupervised learning
- More precisely, semi-supervised learning
- But that is usually not considered too important of a distinction because it is still just learning patterns in the raw data
- But as we mentioned earlier, unlabeled data doesn’t necessarily give the behavior we want

FOUNDATION MODELS

So why use unsupervised learning

- Raw unlabeled data is available in much higher quantity than labeled data
- Unlabeled data is the only data source big enough to train neural nets with billions of weights



When we're learning to see, nobody's telling us what the right answers are — we just look. Every so often, your mother says "that's a dog", but that's very little information. You'd be lucky if you got a few bits of information — even one bit per second — that way. The brain's visual system has $O(10^{14})$ neural connections. And you only live for $O(10^{9})$ seconds. So it's no use learning one bit per second. You need more like $O(10^{5})$ bits per second. And there's only one place you can get that much information: from the input itself.

Geoffrey Hinton
Turing Award co-winner for inventing deep learning

So what do we do?

- We train a base “foundation model” with unsupervised learning
- Then we use supervised techniques to tweak it
- Not unlike a parent teaching their child in Hinton’s quote

FINE TUNING

Fine-Tuning

- In fine-tuning, we start with a foundation model
- Then, we open it up for additional training with some labeled data
  - Typically 50-200 labeled examples is all you need for the neural net to "get it"
- There are a lot of variants of this
  - Leaving the foundation model alone and adding a new layer or two
    - Sometimes called *transfer learning* because it transfers the features of the foundation model to the fine-tuned model
  - Full SGD on the weights in the foundation model
    - Since that may be too expensive, one often only unfreezes a few layers
    - More parameter efficient fine tuning (PEFT) techniques like LoRA and QLoRA exist as well

This seems very abstract Let’s see an example

- Good idea!
- We did the unsupervised base/foundation model building last week
- Let’s look at our first example of fine-tuning
- https://towardsdatascience.com/fine-tune-your-own-llama-2-model-in-a-colab-notebook-df9823a04a32

REINFORCEMENT LEARNING

Reinforcement Learning with Human Feedback (RLHF)

- Humans using an LLM can give positive or negative feedback to the output
  - Either as a pre-release training stage
  - Or by users in production
- That feedback can be used to reward desired behavior and penalize undesired behavior
- This can give fine-grained customization of desired behavior
  - But requires expensive and often skilled manual interaction
- For LLMs, the state is the prompt, and the action is the generated text

HOW DO THESE FIT TOGETHER?

How do these fit together?

- Unsupervised learning is used to train a base/foundation model
- Supervised learning with labeled data is used to fine-tune the model
- Reinforcement learning is used to help the model incorporate feedback on the quality of its results

Yann Lecun’s Layer Cake

Yann Lecun, who also co-won the Turing Award for Inventing Deep Learning, describes theses different levels of learning as a layer cake

How Much Information is the Machine Given during Learning?

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Example: Instruction Tuning

- Generally we want our LLMs to be instruction-tuned
- Meaning they do what they are asked
- E.g., “What is the biggest city in the world?” should be answered, not extended into a questionnaire
- GPT-3 was not instruction-tuned, leading to some of the examples of the original Bing chat veering off from answers
- The GPT-3.5 in the next version of Bing chat was instruction tuned by
  1. Starting with a raw unsupervised/semi-supervised LLM
  2. Fine-tuning it with supervised training examples
  3. RLHF

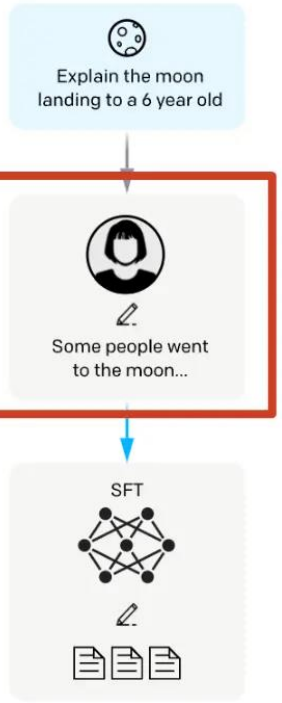
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

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



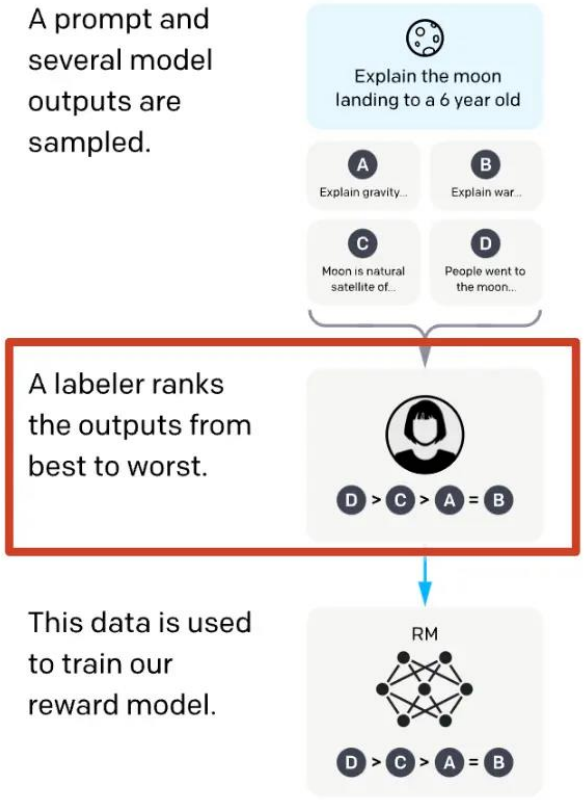
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

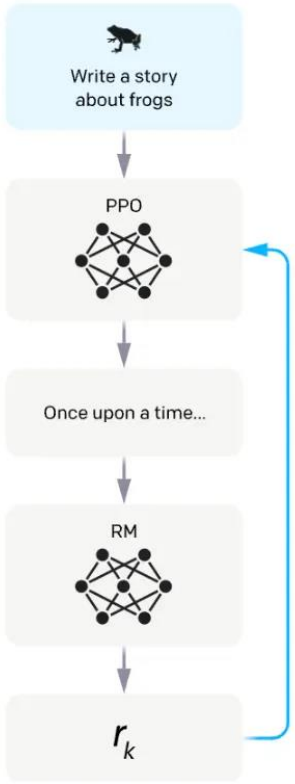
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Let’s take a quick look at an instruction-tuning notebook

- https://towardsdatascience.com/fine-tune-your-own-llama-2-model-in-a-colab-notebook-df9823a04a32

FEW-SHOT LEARNING

Prompt Engineering

Customizing a model sounds hard

- Is there a way we can customize an LLM to our need without having to create a new model?
- Yes!
- Put the information about what you want in the prompt
- This is either called
  - *few-shot learning* because you provided a few “do-like-this” examples
  - *Prompt engineering* because you are tweaking the prompt to elicit a better response
- Prompt engineering is a great example of the democratization of AI because it can be done by anyone, not just techies

Prompt Engineering Example

- The following example was done with Flan-T5
  - Instruction-Finetuned LANguage model
  - https://huggingface.co/docs/transformers/model_doc/flan-t5
- It is based on the excellent Generative AI on Amazon SageMaker – Workshop
  - https://catalog.us-east-1.prod.workshops.aws/workshops/972fd252-36e5-4eed-8608-743e84957f8e/en-US

Zero-shot prompt 😔

- Prompt:
  - Message: I am trying to cook tofu with tomatoes.
    Location:
- Response:
  - San Francisco

One-shot prompt ☺

- Prompt:
  - Indicate the location from the message below

    Message: When the spaceship landed on Mars, the whole humanity was excited
    Location: Space

    Message: I am trying to cook tofu with tomatoes
    Location:
- Response:
  - kitchen

PROMPT-ENGINEERING BEST PRACTICES

Isa Fulford's and Andrew Ng's prompt engineering tactics

- Our rules are from Isa Fulford's and Andrew Ng's short course on prompt engineering
- https://learn.deeplearning.ai/chatgpt-prompt-eng/
- The course is currently free and takes about an hour
- You are encouraged to go through it
- I'll run in Hegel AI's prompt playground
  - https://prompttools.streamlit.app/
  - https://github.com/hegelai/prompttools
  - Created by MPCS graduate Steve Krawczyk and his associates!
- But you can use any of
  - ChatGPT
  - Our Hello-GPT-3 notebooks
  - The notebooks in the deeplearning.ai course
- **Note:** While this deck includes the recommendations, to see the actual prompts, please go to the deeplearning.ai course mentioned above. You can see the prompts in https://learn.deeplearning.ai/chatgpt-prompt-eng/lesson/2/guidelines

USE DELIMITERS TO CLEARLY INDICATE DISTINCT PARTS OF THE INPUT

ASK FOR STRUCTURED OUTPUT

ASK THE MODEL TO CHECK WHETHER CONDITIONS ARE SATISFIED

“FEW SHOT” PROMPTING

SPECIFY THE STEPS REQUIRED TO COMPLETE A TASK

ASK FOR OUTPUT IN A SPECIFIED FORMAT

INSTRUCT THE MODEL TO WORK OUT ITS OWN SOLUTION BEFORE RUSHING TO A CONCLUSION

MODEL LIMITS: HALLUCINATIONS

WHAT DOES THIS MEAN TO ME?

Very few people create their own foundation models

- These large foundation models can cost tens of millions of dollars to train
- So you will most likely choose an existing foundation model and follow one of our customization strategies
- If you do get to work on creating a new foundation model
  - Then I'm jealous☺

Prompt-engineering is most popular

- It’s quick and easy to iterate on
- By comparison, fine-tuning a model on sample prompt-instructions can take minutes and dollars for each iteration

Fine-tuning and RLHF can help scale

- It is not terribly inaccurate to think of fine-tuning as prompt-engineering where you learn the prompt aids once instead of processing them for each question
- OpenAI charges by the token, so this can be a win
- Latency is reduced because it is baked into the model
- It is not unreasonable to start with prompt engineering and then go to fine-tuning when you need to optimize cost at scale
- Totally ok if you want to steer clear of

HOMEWORK

HW 3-1

- Some models support additional pre-training where you can incrementally extend the pretraining
- How do you think that differs from fine-tuning
- What do you think are its benefits?
- What do you think are its disadvantages?

HW 3-2

- We did our prompt engineering examples with GPT 3-5
- Choose another model
  - You can use any free or prepaid model you have access too
  - I will also create a playground with additional prepaid models this weekend
- Do the rules work the same?
- Be ready to report briefly (1-2 minutes) on what you learned in class
- **Note:** To see the actual prompts illustrating the guidelines, go to Lesson 2 of the free deeplearning.ai prompt engineering class at https://learn.deeplearning.ai/chatgpt-prompt-eng/

HW 3-3

- Investigate solving a problem with prompt engineering
- You can either use the following one or choose one of your own
- ChatGPT has a lot of trouble playing Hangman as shown at https://chat.openai.com/share/f677fd11-3b3d-4820-a3ba-cac07de84fed
- Can you use prompt engineering to improve the situation?
  - Show your work. Whether or not you succeed, your grade will be based on what you tried and what conclusions you drew
- I will release a prepaid way for our students to use GPT-4 this weekend
  - Or you can do it on the free GPT-3.5 (or another model)