What We Know About ChatGPT

Robin Jia
USC CSCI 467, Spring 2023
April 25, 2023

Feedback

Questions | Responses 6 | Settings

Any topics you hope we can spend more time on?

2 responses

Further explanation of steps that involve doing math would be helpful for me.

Are we going to talk about contents related to ChatGPT?

ChatGPT

R

What machine learning model should I use to determine if an image contains a leopard or a tiger?

For image classification tasks such as this one, a convolutional neural network (CNN) is typically used. CNNs are particularly good at detecting patterns in visual data and are widely used for image classification tasks.

To train a CNN for your specific task, you would first need to gather a dataset of images containing leopards, tigers, and possibly other animals. This dataset would be split into training, validation, and test sets, with the training set used to train the model, the validation set used to tune the model's hyperparameters, and the test set used to evaluate the model's performance on unseen data.

There are many pre-trained CNN models that you can use as a starting point, such as ResNet, VGG, or Inception. You can also choose to train your own model from scratch using a framework such as TensorFlow or PyTorch.

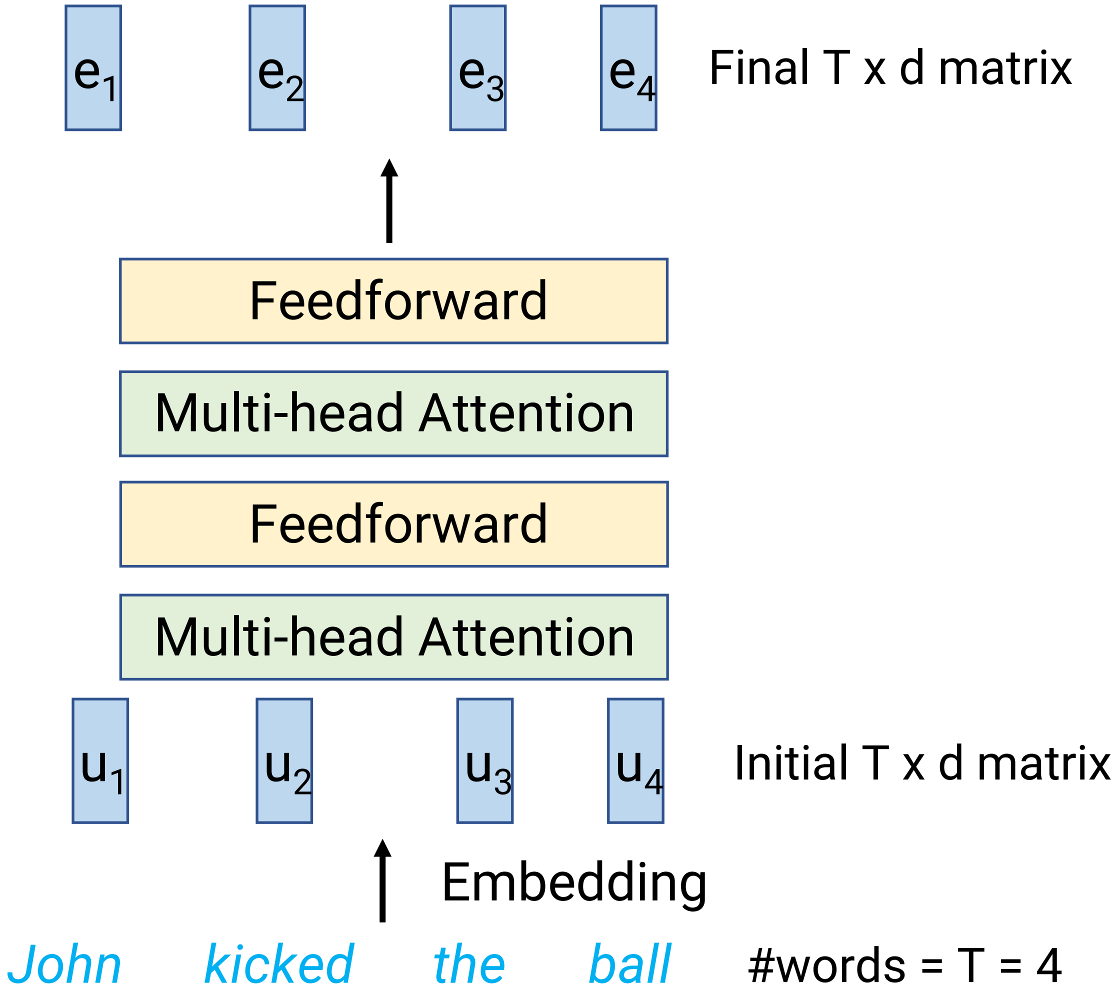
Once you have a trained model, you can pass in a new image and the model will output a probability distribution over the possible classes, in this case, leopard or tiger. The class with the highest probability would be the predicted class for the image.

- Language model-based Chatbot developed by OpenAI
- Large (but not largest) Transformer model
- Lots of information not public, but we do know some things…

Outline

- Technical ingredients of ChatGPT
    - Language model pretraining
    - Reinforcement Learning from Human Feedback
- Limitations and concerns

Previously: Transformers



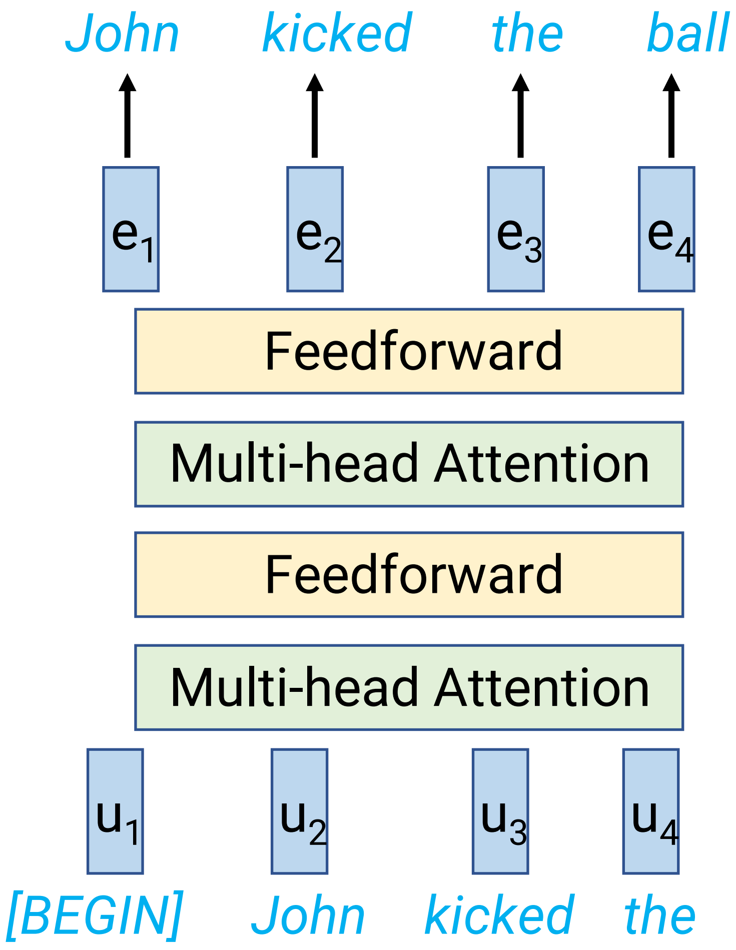
- One transformer consists of
  - Embeddings for each word of size d
    - Let T =#words, so initially T x d matrix
  - Alternating layers of
    - "Multi-headed" attention layer
    - Feedforward layer
    - Both take in T x d matrix and output a new T x d matrix
  - Plus some bells and whistles
    - Residual connections & LayerNorm
    - Byte pair encoding tokenization

Autoregressive Language Model Training

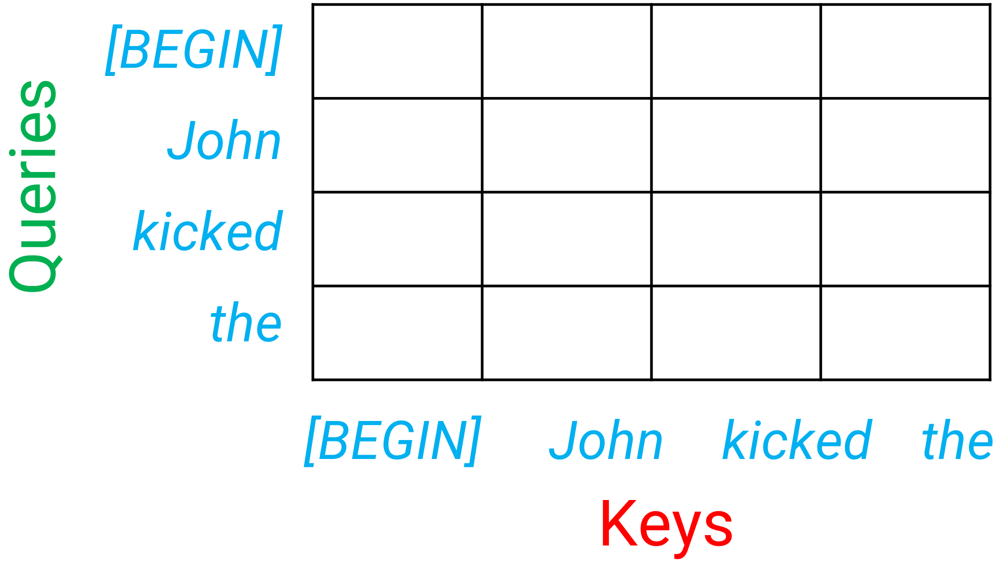
- Training example: “Convolutional neural networks are good for image classification”
- Want to maximize P(“Convolutional neural networks are good for image classification”)
- Take log and decompose by chain rule:

  log P(“Convolutional”)
  + log P(“neural” | “Convolutional”)
  + log P(“networks” | “Convolutional neural”)
  + log P(“are” | “Convolutional neural networks”) + …
- Decomposes into a bunch of **next-word-classification** problems
  - I will also write this as P(word | prefix)

Review: Transformer autoregressive decoders



- How to do autoregressive language modeling?
- Test-time
  - At time t, attend to positions 1 through t
  - Happens in series



Review: Transformer autoregressive decoders

- How to do autoregressive language modeling?
- Training time: Masked attention trick
  - Recall: Attention computes Q x K^T (T x T matrix), then does softmax
  - But if generating autoregressively, time t can only attend to times 1 through t
  - Solution: Overwrite Q x K^T to be $-\infty$ when query index < key index
  - **All timesteps happen in parallel**

Queries (rows) / Keys (columns)

	[BEGIN]	*John*	*kicked*	*the*
[BEGIN]	10	-2	6	3
John	0	7	2	-4
kicked	-3	4	5	-8
the	2	1	7	6

Review: Transformer autoregressive decoders

- How to do autoregressive language modeling?
- Training time: Masked attention trick
  - Recall: Attention computes Q x K^T (T x T matrix), then does softmax
  - But if generating autoregressively, time t can only attend to times 1 through t
  - Solution: Overwrite Q x K^T to be $-\infty$ when query index < key index
  - **All timesteps happen in parallel**

Queries (rows) / Keys (columns)

	[BEGIN]	John	kicked	the
[BEGIN]	10	-2	6	3
John	0	7	2	-4
kicked	-3	4	5	-8
the	2	1	7	6

Review: Transformer autoregressive decoders

- How to do autoregressive language modeling?
- Training time: Masked attention trick
  - Recall: Attention computes Q x K^T (T x T matrix), then does softmax
  - But if generating autoregressively, time t can only attend to times 1 through t
  - Solution: Overwrite Q x K^T to be $-\infty$ when query index < key index
  - **All timesteps happen in parallel**

Queries \ Keys	*[BEGIN]*	*John*	*kicked*	*the*
[BEGIN]	10	$-\infty$	$-\infty$	$-\infty$
John	0	7	$-\infty$	$-\infty$
kicked	-3	4	5	$-\infty$
the	2	1	7	6

Pre-trained language models

- GPT-3 (2020): A 175 billion parameter language model
  - Architecture
    - 96 Transformer layers
    - 12288-dimensional hidden states
    - 96 heads in each attention layer
  - Trained on a very large corpus of documents scraped from the web
    - Some filters used to promote data quality
    - One strategy: Train classifier to distinguish random internet documents from ones from known “high-quality” sources, drop documents with low classifier score
- ChatGPT: Reportedly 20 billion parameters
  - Easier to deploy at scale than 175B model
  - Likely was first pretrained in a similar manner as GPT-3
  - But then an additional training step was added…

https://arxiv.org/pdf/2005.14165.pdf

Outline

- Technical ingredients of ChatGPT
  - Language model pretraining
  - Reinforcement Learning from Human Feedback
- Limitations and concerns

Supervised fine-tuning

- Pretraining stage: Train on all the data you can get access to
  - Pros: A lot of data avoids overfitting, model can learn about all sorts of long-tail knowledge
  - Cons: You're training the model to imitate the average internet post
  - High quantity, low quality!
- Solution: Fine-tune on a smaller, highly curated dataset after
  - These are examples you really do want the model to imitate
  - Pre-training has taught the model many things; fine-tuning tells it to only verbalize the parts that are desirable

Limitations of Supervised fine-tuning

- Problem 1: Data scale
  - High-quality data is expensive to obtain, you don’t have that much of it (relative to pretraining data)
- Problem 2: Exposure bias

Exposure bias

- Training time: Model learns to predict next word given human-written prefix
- Test time: Model must predict next word given model-written prefix
- **Exposure bias**: Model was never "exposed" to its own outputs during training, so it may not know what to do!

Training: Human-written

P(word | prefix)

Test time: Generated by model

Exposure bias and reinforcement learning

- We can view sequence generation as a reinforcement learning problem!
  - Action: Which word to generate next
  - State: Sequence of all words generated so far
  - Reward: Whether the final complete output is “good”
    - Rewards are 0 for intermediate timesteps
    - Only get non-zero reward at final timestep
- In RL, supervised fine-tuning is called “**imitation learning**”
  - Known to be suboptimal due to exposure bias
  - You can try to mimic an expert player, but a worse player also needs to know how to recover from mistakes

Limitations of Supervised fine-tuning

- Problem 1: Data scale
  - You simply don’t have much data at your disposal
- Problem 2: Exposure bias
- **Problem 3: Can promote guessing rather than factual responses**

Review: Neural networks as feature learners

Input x

Classifier 1:
Is front clear?

Classifier 2:
Is left clear?

Classifier 3:
Is right clear?

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Classifier 4:
Where to go?

Output y
Turn left

<u>**Learn a classifier whose output is a good feature**</u>
We don’t tell the model what classifier to learn
Model must learn that “is front clear” is a useful concept

Learn to classify based on features
(same as linear model)

Supervised Fine-tuning and Factuality

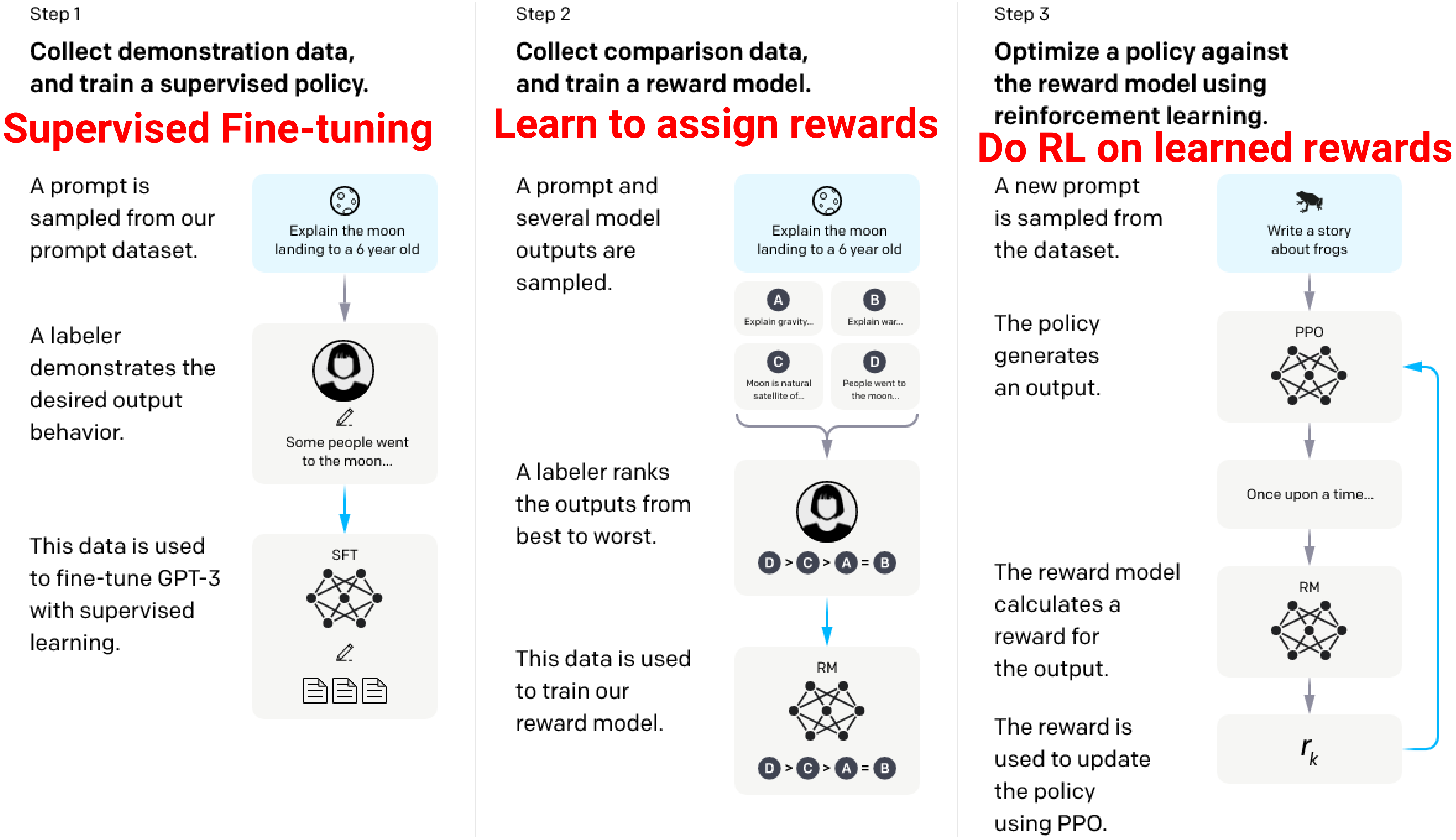
- Consider the following fine-tuning examples
  - Prompt: "When was the US Declaration of Independence signed?"
    Answer: "July 4, 1776"
    - Model has probably seen this information many times during pre-training
    - So it has probably learned features that associate the Declaration of Independence with July 4
  - Prompt: "When was Robin Jia born?"
    Answer: "[…]"
    - Model has probably (?) not seen this information during pre-training
    - Cannot have learned features associating me with my birthday
    - Supervised fine-tuning **encourages the model to just make something up**!

See: John Schulman's talk https://www.youtube.com/live/hhiLw5Q_UFg?feature=share

Limitations of Supervised fine-tuning

- Problem 1: Data scale
  - You simply don't have much data at your disposal
- Problem 2: Exposure bias
- Problem 3: Can promote guessing rather than factual responses
- Solution: **Fine-tune the language model with reinforcement learning!**
  - Use a reward that encourages the **model's full outputs** to be **correct/factual**
  - Reward will be **computed with another model** (can get infinite data now)

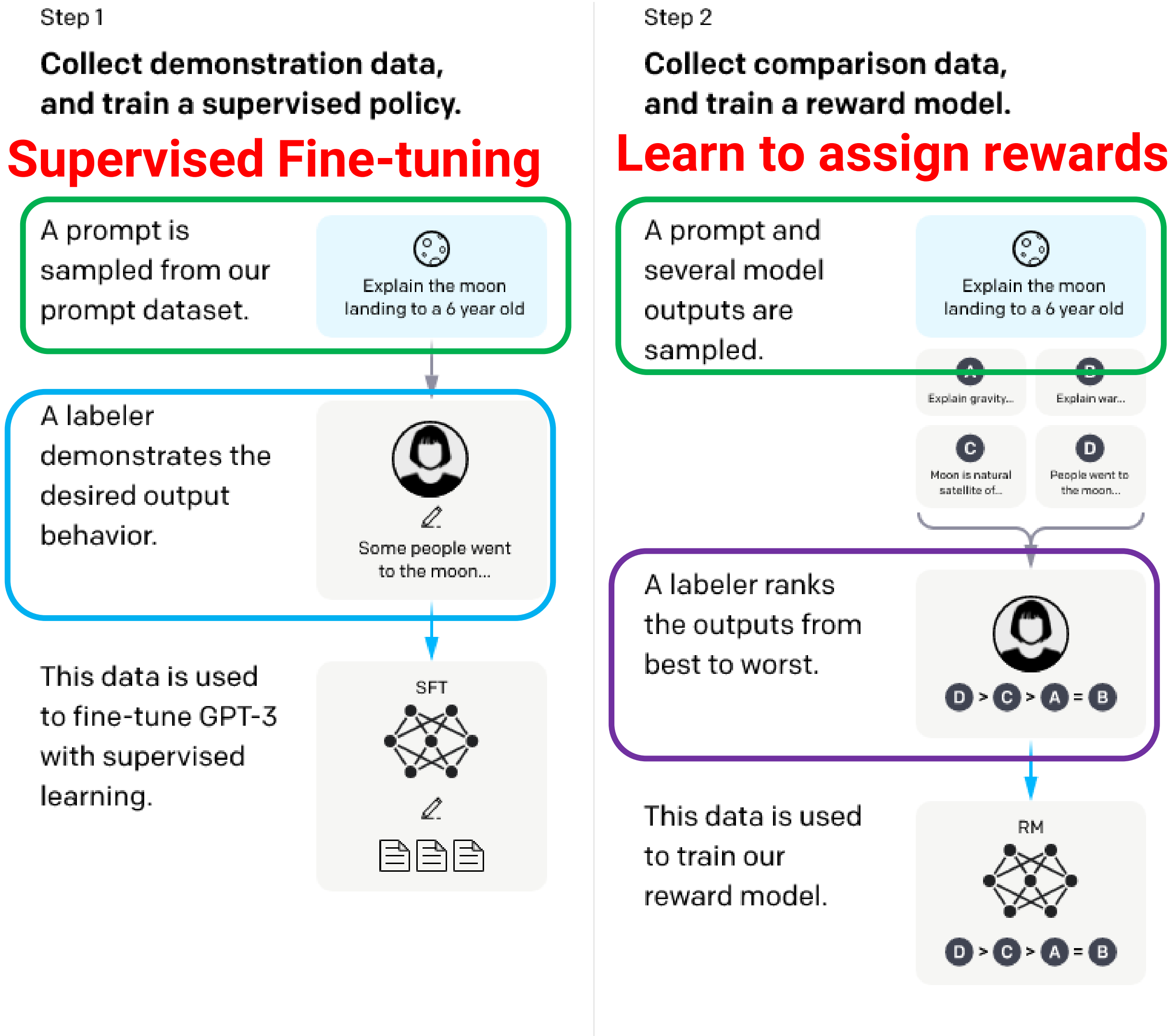
Reinforcement Learning from Human Feedback



- “RLHF” for short
- Trains language model with RL
- Rewards come from a model trained to predict human preferences

https://openai.com/research/instruction-following

RLHF: The Data



- Part 1: Prompts
  - Some written by hired annotators
  - Some based on use-cases from waitlist applications to OpenAI (!)
  - Some from actual user queries to the OpenAI API
- Part 2: Demonstrations
  - Hired annotator writes a desired response to prompt
- Part 3: Rankings
  - Sample several model responses from model's probability distribution
  - Hired annotator ranks them from best to worst

RLHF: Who's behind the data?

- InstructGPT paper (precursor to ChatGPT): "We hired a team of about 40 contractors on Upwork and through ScaleAI"
- Labelers had to pass various screening tests

What gender do you identify as?	
Male	50.0%
Female	44.4%
Nonbinary / other	5.6%
What ethnicities do you identify as?	
White / Caucasian	31.6%
Southeast Asian	52.6%
Indigenous / Native American / Alaskan Native	0.0%
East Asian	5.3%
Middle Eastern	0.0%
Latinx	15.8%
Black / of African descent	10.5%
What is your nationality?	
Filipino	22%
Bangladeshi	22%
American	17%
Albanian	5%
Brazilian	5%
Canadian	5%
Colombian	5%
Indian	5%
Uruguayan	5%
Zimbabwean	5%

What is your age?	
18-24	26.3%
25-34	47.4%
35-44	10.5%
45-54	10.5%
55-64	5.3%
65+	0%
What is your highest attained level of education?	
Less than high school degree	0%
High school degree	10.5%
Undergraduate degree	52.6%
Master's degree	36.8%
Doctorate degree	0%

https://arxiv.org/pdf/2203.02155.pdf

RLHF: Who’s behind the data?

TIME

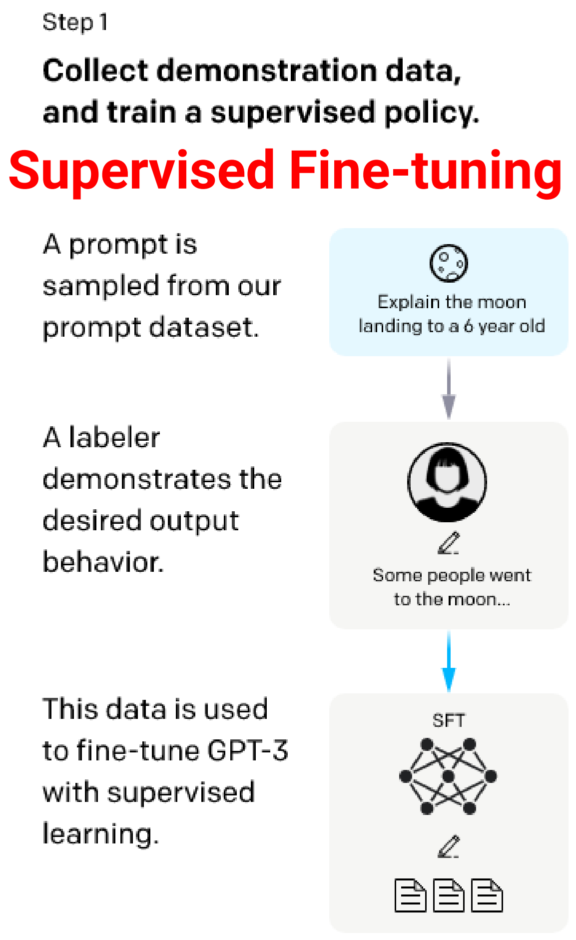
BUSINESS • TECHNOLOGY

Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic

- *“The data labelers employed by Sama on behalf of OpenAI were paid a take-home wage of between around $1.32 and $2 per hour”*
  - Sama hires workers from Kenya, Uganda, and India
- *“OpenAI sent tens of thousands of snippets of text to an outsourcing firm in Kenya, beginning in November 2021. Much of that text appeared to have been pulled from the darkest recesses of the internet. Some of it described situations in graphic detail like [omitted]…The work’s traumatic nature eventually led Sama to cancel all its work for OpenAI in February 2022, eight months earlier than planned.”*

https://time.com/6247678/openai-chatgpt-kenya-workers/

RLHF: Initial Supervised Fine-tuning



- Run imitation learning on labeler-provided demonstrations, given prompt as prefix
- Very similar to pre-training, except:
  - Only compute loss on response tokens, not on prompt tokens
  - Much smaller dataset

Loss on this example =
 log P("Some" | "Explain the moon landing to a 6 year old:")
+ log P("people" | "Explain the moon landing to a 6 year old: Some")
+ ...

RLHF: The Reward Model

Step 2

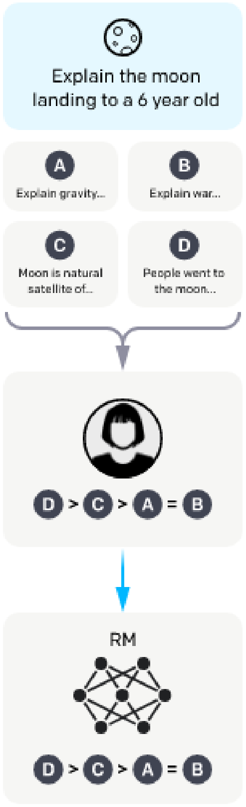
Collect comparison data, and train a reward model.

Learn to assign rewards

A prompt and several model outputs are sampled.

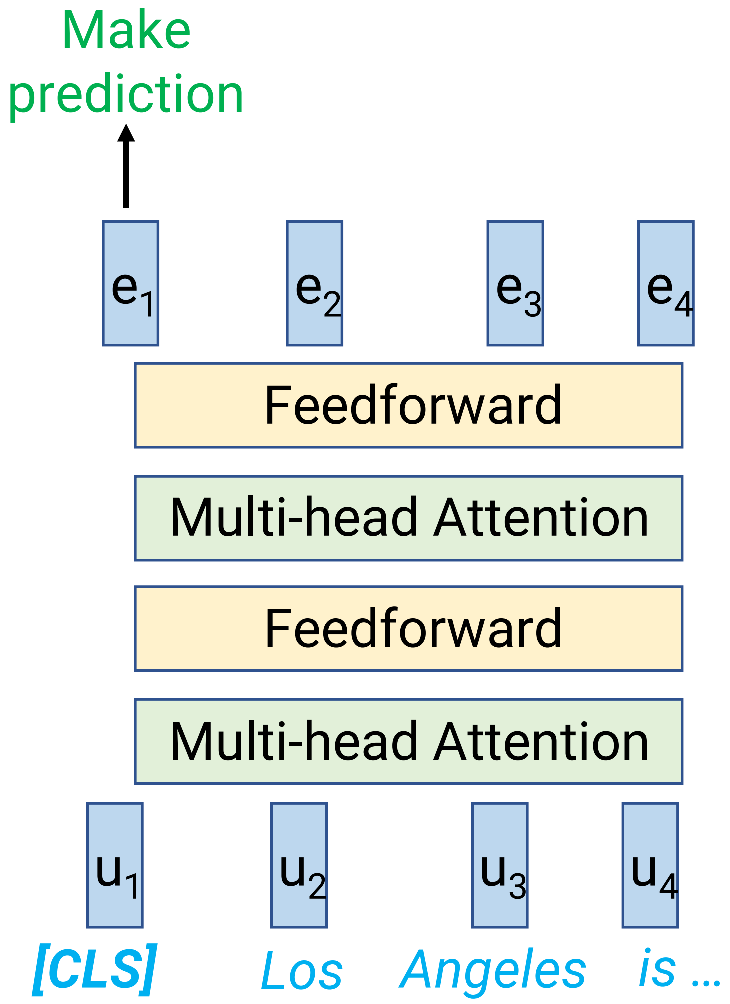
A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



- Goal: Fine-tune ChatGPT with RL
- Step 1: Show human annotators several sampled model outputs, ask them to rank
  - Provides some RL training data, but not a ton
- Step 2: Train a "**reward model**" to predict the human's rankings
  - Now we can run as many RL training episodes as we want for free, using the reward model in place of the human annotators

Previously: BERT Fine-tuning



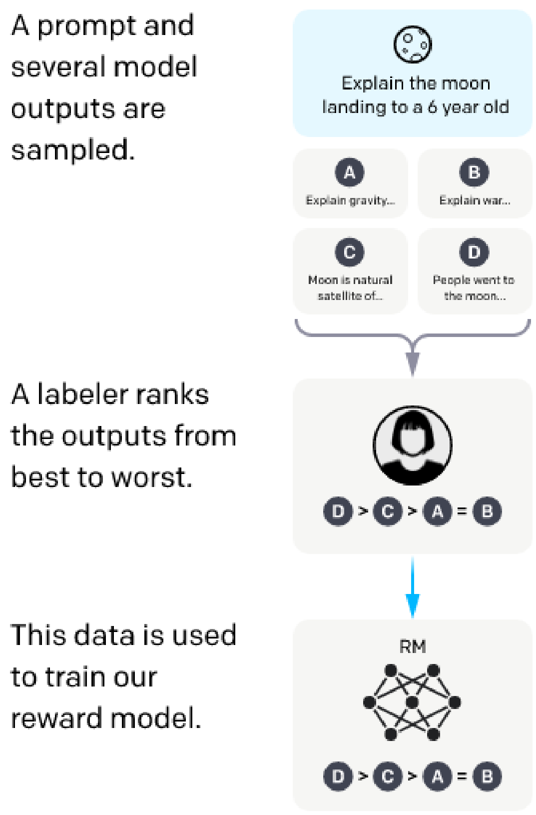
- BERT: Model pre-trained by masked language modeling
- Initialize parameters with BERT
  - BERT was trained to expect every input to start with a special token called [CLS]
- Add parameters that take in the output at the [CLS] position and make prediction
- Keep training all parameters ("fine-tune") on the new task
- Reward model is a similar model that was pretrained, then fine-tuned to predict reward

RLHF: The Reward Model

Step 2

Collect comparison data, and train a reward model.

Learn to assign rewards



- Model
  - 6 billion parameter pretrained language model
  - Then fine-tune all parameters (like BERT)
  - A smaller model than ChatGPT itself

RLHF: Training the Reward Model

Step 2

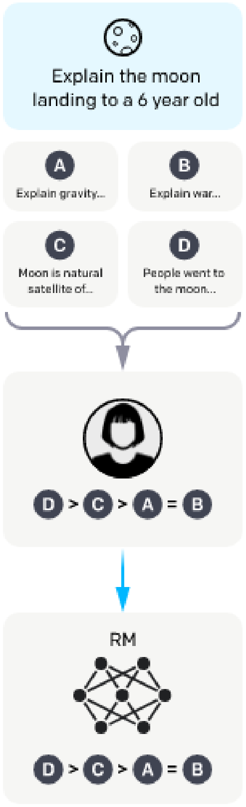
Collect comparison data, and train a reward model.

Learn to assign rewards

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.

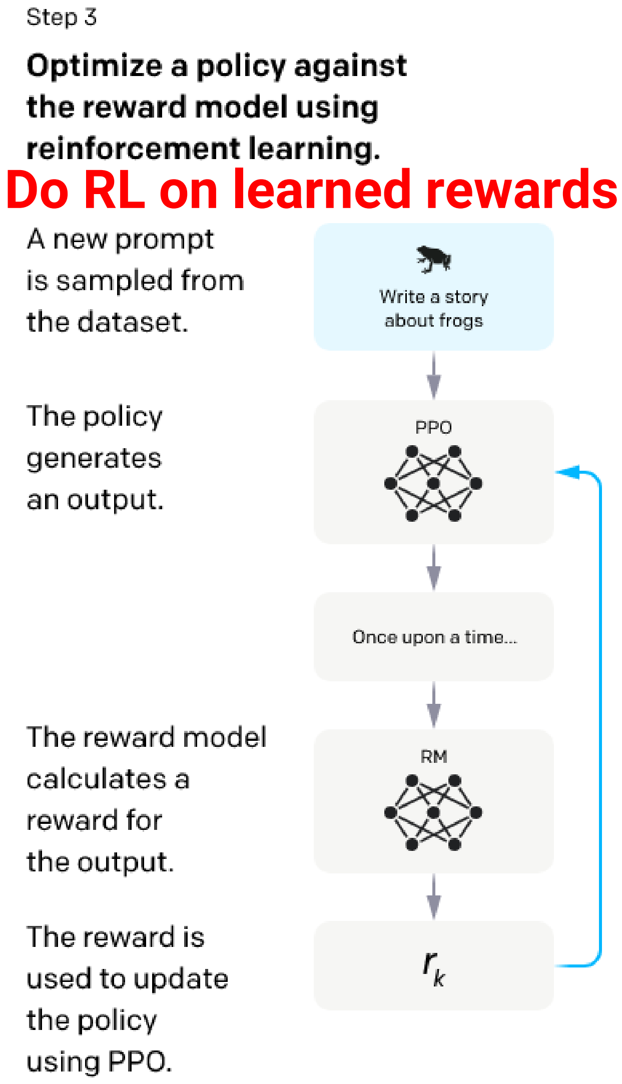


- Input: Prompt x, winning output y_w, losing output y_l
  - Wins/losses determined by labelers' rankings
- Reward model predicts scalar reward $r_\theta(x, y)$ given prompt x and model output y
- Objective on one example is to minimize:

$$-\log \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))$$

  - Want reward on y_w to be higher than on y_l
  - Use the familiar logistic regression loss function!
    - Loss goes to 0 if argument is large
    - Loss goes to infinity if argument is small
  - Binary classification of which output is better

RLHF: Doing RL on the Language Model



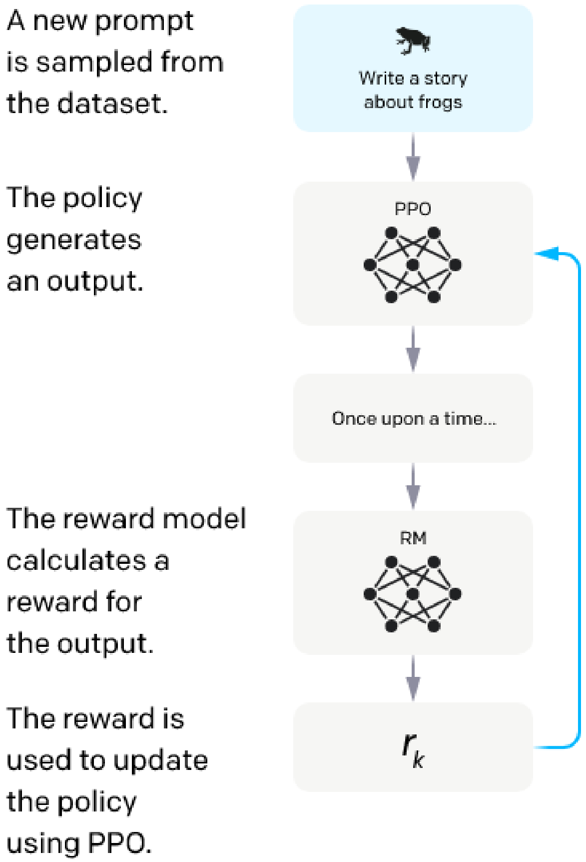
- Which RL algorithm should we use?
  - Deep Q-Learning?
  - Policy Gradient?
- To answer this, we should ask: How does a language model define a RL policy?
  - It directly classifies the next action (i.e., next word) in the current state (i.e., sequence of words generated so far)
    - This is what **policy gradient** requires!
  - It does not predict a Q-value, like Q-learning expects

RLHF: Doing RL on the Language Model

Step 3

Optimize a policy against the reward model using reinforcement learning.

Do RL on learned rewards



- We need to use policy gradient!
  - Recall: Policy gradient computes a quantity whose expected value is the gradient w.r.t. parameters of the expected reward, then uses that quantity to do gradient ascent
- Algorithm of choice: Proximal Policy Optimization (PPO)
  - Basic policy gradient method estimates the gradient of expected reward, but with very high variance
  - Idea 1: Use “advantage” (how much this action improves reward vs. baseline) instead of raw rewards—lowers variance
  - Idea 2: Only make small updates to the policy at each step, in case the estimated gradient goes in the wrong direction

RLHF and Factuality

- Let’s revisit the earlier examples:
  - Prompt: “When was the US Declaration of Independence signed?”
    Answer: “July 4, 1776”
  - Model outputs
    - “July 4, 1776”
    - “January 1, 1950”
    - “I don’t know”

  - Top-ranked answer is the real answer

See: John Schulman’s talk https://www.youtube.com/live/hhiLw5Q_UFg?feature=share

RLHF and Factuality

- Let’s revisit the earlier examples:
  - Prompt: “When was Robin Jia born?”
    Answer: “[…]”
  - Model outputs
    - “December 7, 1831”
    - “May 23, 1989”
    - “I don’t know”

  - Top-ranked answer is to say “I don’t know”
- Overall effect: Model encouraged to say “I don’t know” when its top candidate outputs are wrong

See: John Schulman’s talk https://www.youtube.com/live/hhiLw5Q_UFg?feature=share

Announcements

- Homework 4 due this Thursday
- No section on Friday
- Final Exam May 4, 2-4pm
  - Can bring two sheets of paper with notes, no other assistance
- Final project reports due May 9
  - You will be asked to submit code & commands that generate your results (will update website shortly)
- Regrade Requests
  - Last day for regrade requests from HW0/HW1/HW2/Midterm May 1
  - Last day for any regrade requests will be May 9
  - Can't accommodate regrade requests on final project reports (need to finalize grades)

Outline

- Technical ingredients of ChatGPT
  - Language model pretraining
  - Reinforcement Learning from Human Feedback
- Limitations and concerns

Google

who was the king of england during the american revolution?

All Images News Videos Maps More Settings Tools

About 57,300,000 results (0.90 seconds)

George

George III (1738-1820) **George** III was the King of Great Britain and Ireland during the American Revolution. The death of his father, Frederick Lewis, the Prince of Wales, in 1751 meant that the 22-year-old prince succeeded his grandfather, **George** II, to the throne in 1760.

George III - The American Revolution

www.ouramericanrevolution.org › index.cfm › people › view

People also search for

View 15+ more

Google

who was the king of portugal during the american revolution?

All News Images Videos Shopping More Settings Tools

About 17,300,000 results (0.93 seconds)

John VI

John VI, (born May 13, 1767, Lisbon, **Portugal**—died March 10, 1826, Lisbon), prince regent of **Portugal** from 1799 to 1816 **and king** from 1816 to 1826, whose reign saw the **revolutionary** struggle **in** France, the Napoleonic invasion of **Portugal** (**during** which he established his court **in** Brazil), **and** the implantation of ...

John VI | king of Portugal | Britannica

https://www.britannica.com › biography › John-VI-king-of-Portugal

People also search for

View 15+ more

Google

who was the king of italy during the american revolution?

All | News | Images | Shopping | Videos | More | Tools

About 17,500,000 results (0.81 seconds)

For the remainder of the war, **Victor Emmanuel** lived at Villa Maria Pia at Posillipo. He stayed out of the public eye and, according to his diary, was visited by his son only three times. Following the Allied victory and liberation of northern Italy, the Italian government started to put itself back together. Jul 14, 2021

https://www.nationalww2museum.org › war › articles › fa...

The Fascist King: Victor Emmanuel III of Italy | New Orleans

About featured snippets • Feedback

ARTICLE

The Fascist King: Victor Emmanuel III of Italy

The life of the king $\boxed{\text{who ruled during both World War I and World War II}}$ and participated in the rise and fall of Italian Fascism.

Testing ChatGPT

R who was the king of portugal during the american revolution?

The king of Portugal during the American Revolution was D. José I. He ruled from 1750 until his death in 1777. Portugal did not directly participate in the American Revolution, but it did have a significant impact on the war through its trade relations with Britain and its allies.

R who was the king of italy during the american revolution?

The Kingdom of Italy did not exist during the American Revolution, as Italy was not yet unified as a single country. Instead, the Italian peninsula was divided into various states and territories, each with its own ruler. Some of the prominent states in Italy at that time included the Kingdom of Sardinia, the Republic of Venice, the Grand Duchy of Tuscany, and the Papal States, among others.

Therefore, there was no single king of Italy during the American Revolution.

Testing ChatGPT

R Who was the king of Portugal when Abraham Lincoln was born?

The king of Portugal when Abraham Lincoln was born on February 12 1809, was John VI (João VI in Portuguese). He was the King of Portugal and the Algarves from 1816 until his death in 1826.

R Who was the king of Portugal in 1809?

The king of Portugal in 1809 was Dom João VI.

R Who was the King of Portugal in 1780?

The King of Portugal in 1780 was Dom José I, who had been reigning since 1750.

Joseph I *The Reformer* Portuguese: *José I*	31 July 1750	24 February 1777
Maria I *The Pious; The Mad* Portuguese: *Maria I*	24 February 1777	20 March 1816
Peter III *The Capacidónio; The Builder; The Edifier; The Sacristan; The Enabler* Portuguese: *Pedro III*	24 February 1777	25 May 1786
John VI *The Clement; The Merciful* Portuguese: *João VI*	20 March 1816	10 March 1826

ChatGPT "reasoning"

R

What weighs more, a pound of feathers of a ton of bricks?

Both a pound of feathers and a ton of bricks weigh the same amount, which is 2000 pounds.

This is because a pound is a unit of weight measurement, just like a ton. However, a pound is a much smaller unit than a ton, so it takes a much larger quantity of pounds to equal a ton.

Therefore, while the feathers may take up more space than the bricks, they still weigh the same amount.

- ChatGPT has likely seen this riddle before about a **pound** of feathers vs. **pound** of bricks
- Overgeneralizes based on that answer

ChatGPT can “Hallucinate”

R

Find me a research paper that tests question answering systems with paragraphs that have an extra sentence added at the end

Here's a research paper that fits your request:

Does not exist

Title: Evaluating the Robustness of Question Answering Systems to Sentence-Level Perturbations

Authors: Prawira Saskara, Daya Guo, Anh Tuan Luu, Jinho D. Choi

Not real(?) | NLP PhD Student | NLP Prof. (NTU) | NLP Prof. (Emory)

Journal: Pr… Computational Linguistics

Link to the paper: https://www.aclweb.org/anthology/P19-1148.pdf

URL links to:

Exact Hard Monotonic Attention for Character-Level Transduction

Shijie Wu[a] and **Ryan Cotterell**[a,fi]
[a]Department of Computer Science, Johns Hopkins University
[fi]Department of Computer Science and Technology, University of Cambridge
shijie.wu@jhu.edu, rdc42@cam.ac.uk

Summary: The paper presents an evaluation framework for testing the robustness of question answering systems to sentence-level perturbations. Specifically, the authors introduce a new dataset called SQuAD-AddSent, which consists of the original SQuAD paragraphs with an extra sentence added at the end. The added sentence can either be a

- This is a description of one of my most cited research papers from 2017
- We did actually have a method called AddSent and applied it to the SQuAD dataset

Consequences of "Hallucinations"

REUTERS® World Business Markets Legal More

Technology

3 minute read · April 5, 2023 11:52 AM PDT · Last Updated 19 days ago

Australian mayor readies world's first defamation lawsuit over ChatGPT content

By Byron Kaye

"Brian Hood, who was elected mayor of Hepburn Shire, 120km (75 miles) northwest of Melbourne, last November, became concerned about his reputation when members of the public told him ***ChatGPT had falsely named him as a guilty party in a foreign bribery scandal*** *involving a subsidiary of the Reserve Bank of Australia in the early 2000s.*

Hood did work for the subsidiary, Note Printing Australia, but ***was the person who notified authorities about payment of bribes*** *to foreign officials to win currency printing contracts, and was* ***never charged with a crime****, lawyers representing him said."*

https://www.reuters.com/technology/australian-mayor-readies-worlds-first-defamation-lawsuit-over-chatgpt-content-2023-04-05/

Conclusion

- How does ChatGPT work?
  - Stage 1: Pre-training on large corpus of text
  - Stage 2: RLHF
    - Supervised fine-tuning on human demonstrations
    - Train a reward model to provide feedback to LM
    - Fine-tune LM with policy gradient (PPO) to maximize rewards given by reward model
- Words of caution
  - ChatGPT answers may be made up!
  - Useful for brainstorming and suggestions, bad for facts
  - Likelihood of success depends on commonality of data in pre-training/RLHF datasets