ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

TRY CHATGPT ↗

November 30, 2022
13 minute read

ChatGPT

From Transformers to Reinforcement Learning from Human Feedback (RLHF)

Presented by:
John Tan Chong Min

Combination of both rule-based and Neural Networks

- Neural Network:
  - Transformers for next-token generation
  - Reward Model for reinforcement learning

- Rule-based: Moderation API

Transformers

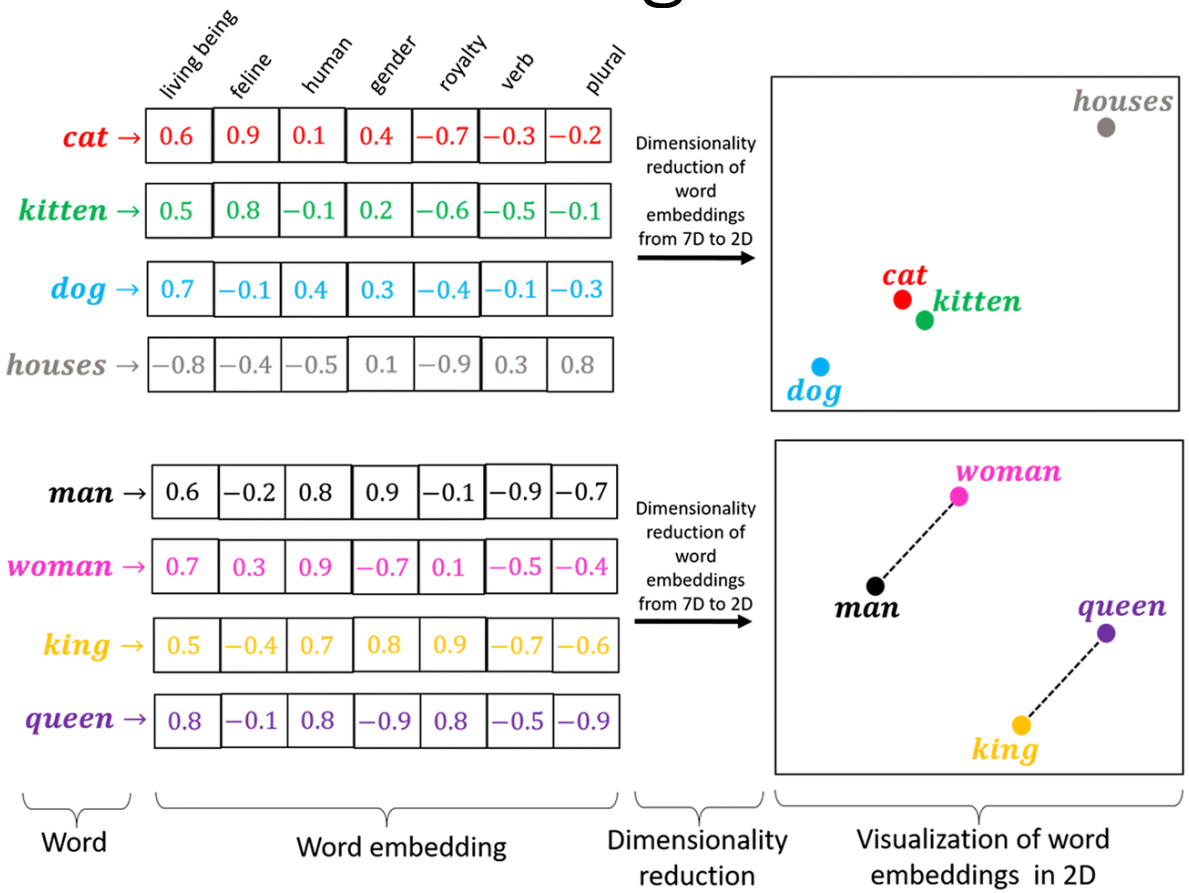
Attention Is All You Need. 2017. Vaswani et al.

Improving language understanding by generative pre-training. 2018. Radford et al. (GPT paper)

Embedding Space

Semantic Meaning

Word Embeddings



- Extracting semantic meaning in higher-dimensional space
- Number of dimensions depends on use case

Taken from: https://medium.com/@hari4om/word-embedding-d816f643140

Transformer Architecture

Summarized with illustrations from

https://jalammar.github.io/illustrated-transformer/

Transformers: Overall

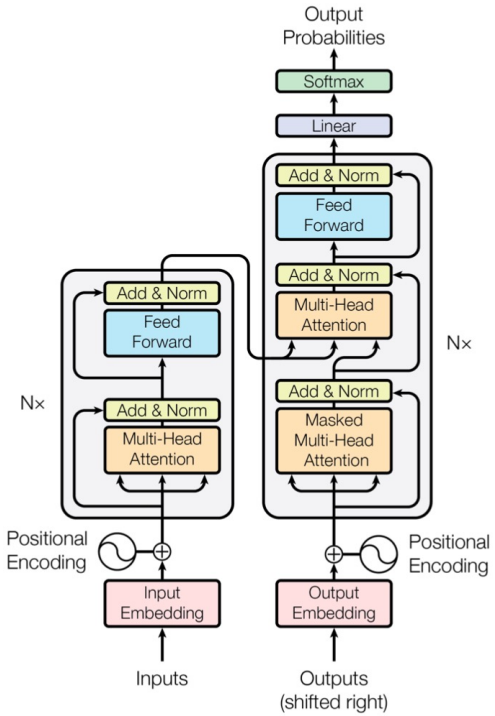


Figure 1: The Transformer - model architecture.

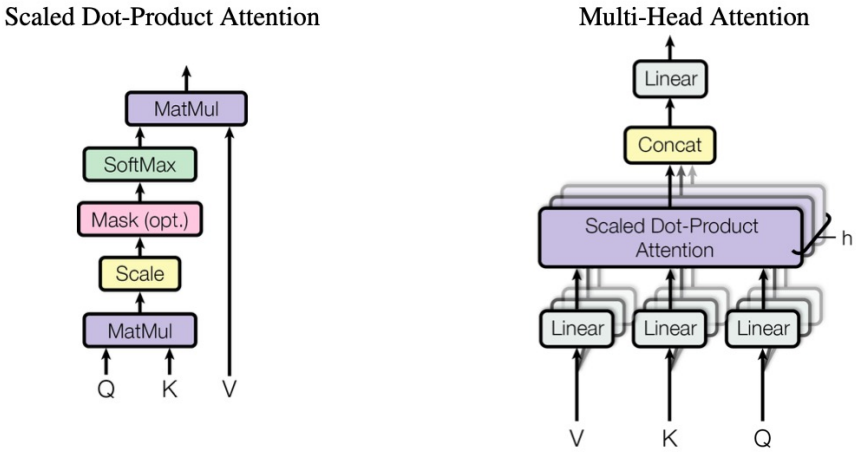


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Taken from: Attention is all you need. Vaswani et al. (2017)

Encoder-Decoder link

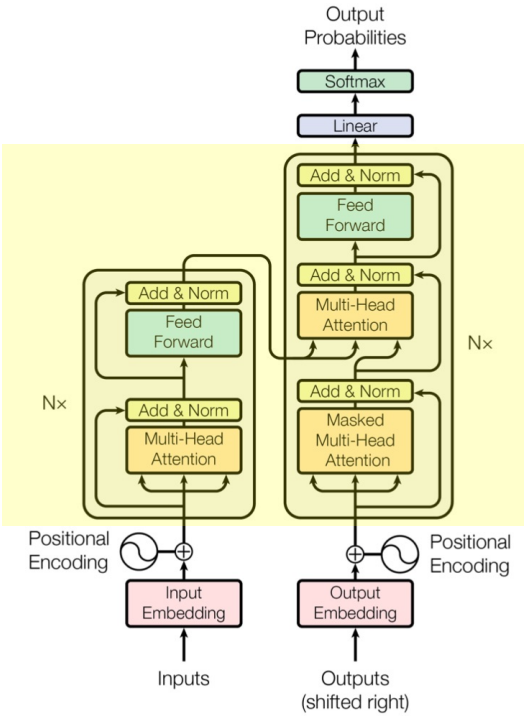
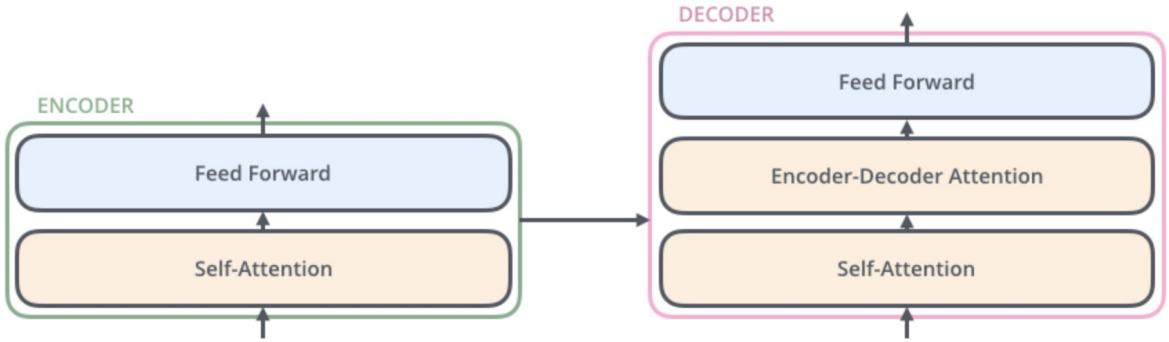


Figure 1: The Transformer - model architecture.

Self-attention in the Encoder & Decoder block,
Then Encoder-Decoder Attention



https://jalammar.github.io/illustrated-transformer/

Self-Attention block

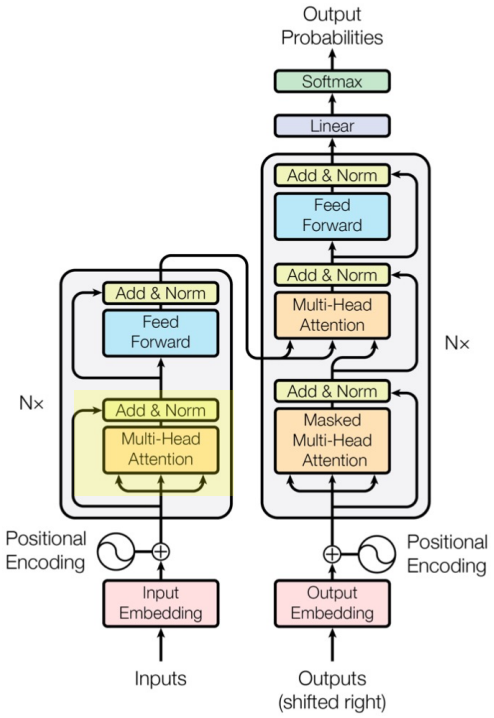
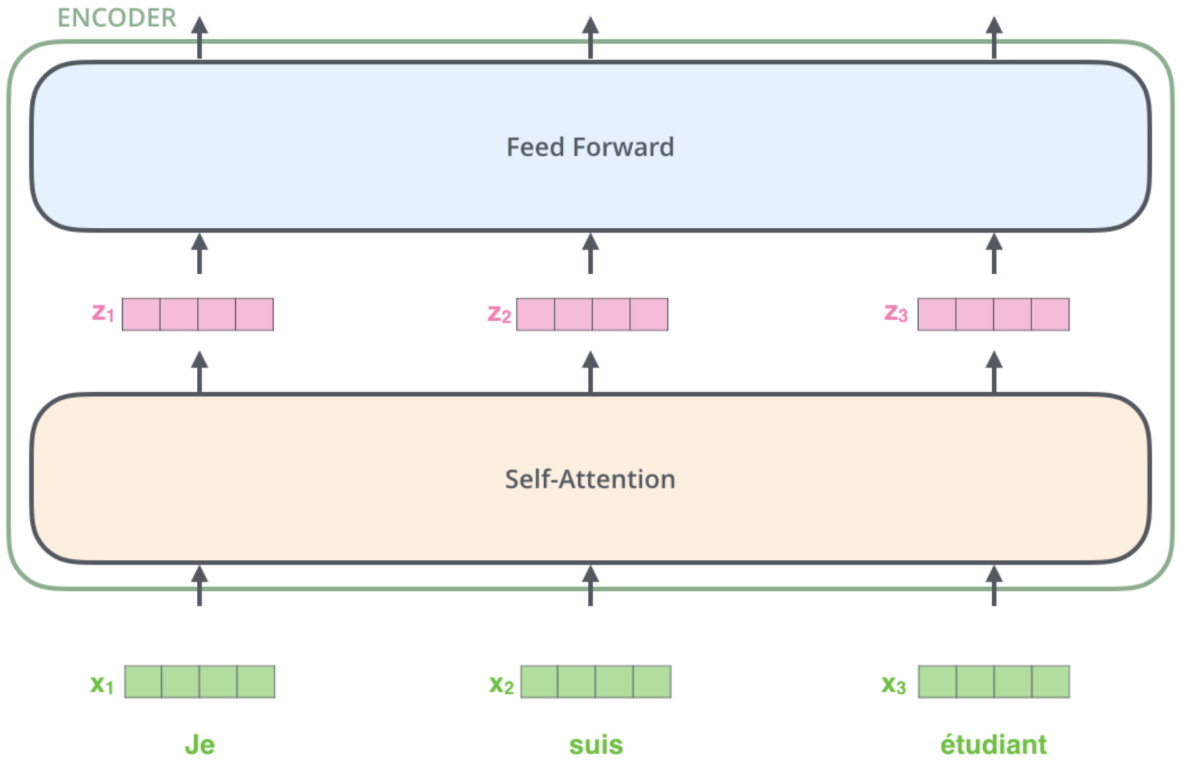


Figure 1: The Transformer - model architecture.

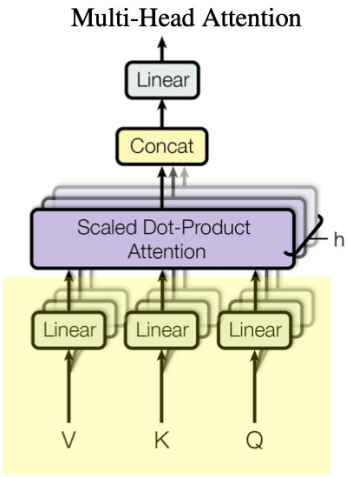
Converting input dimension to embedding dimensions

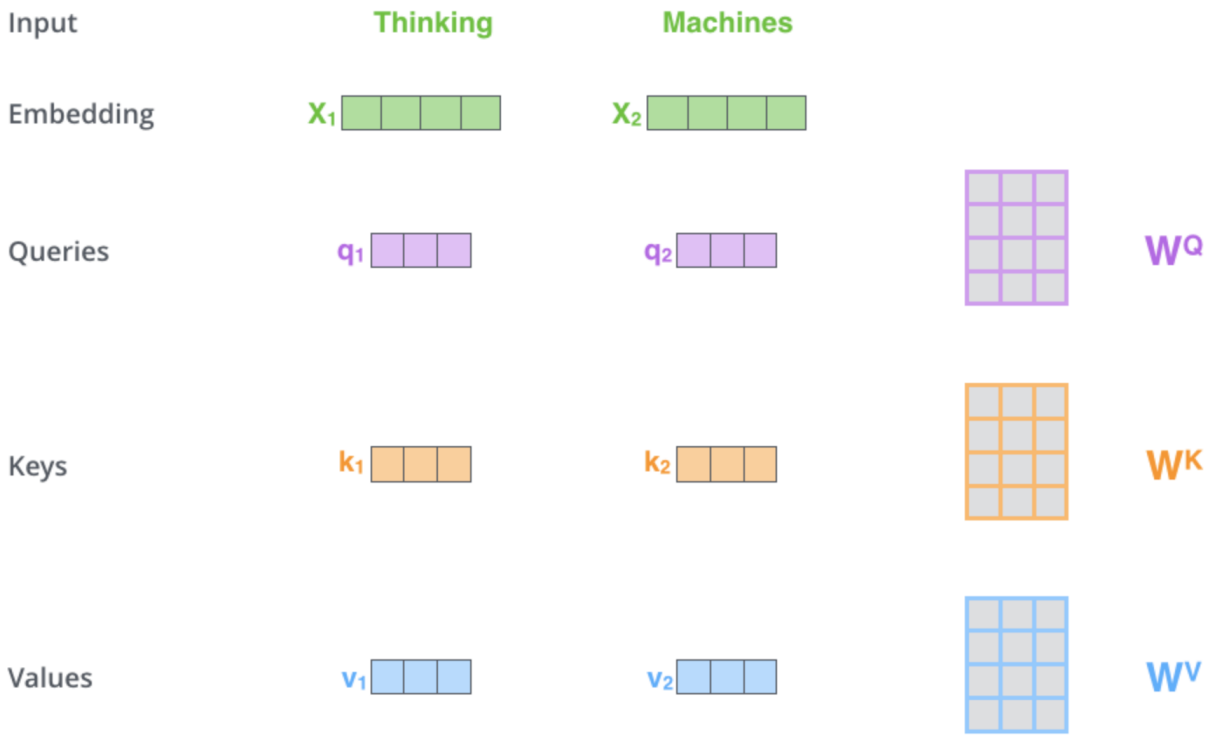


https://jalammar.github.io/illustrated-transformer/

Generate Embedding Space

Convert Q, K, V to embedding space dimensions





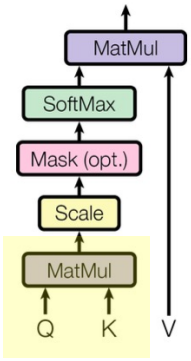
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

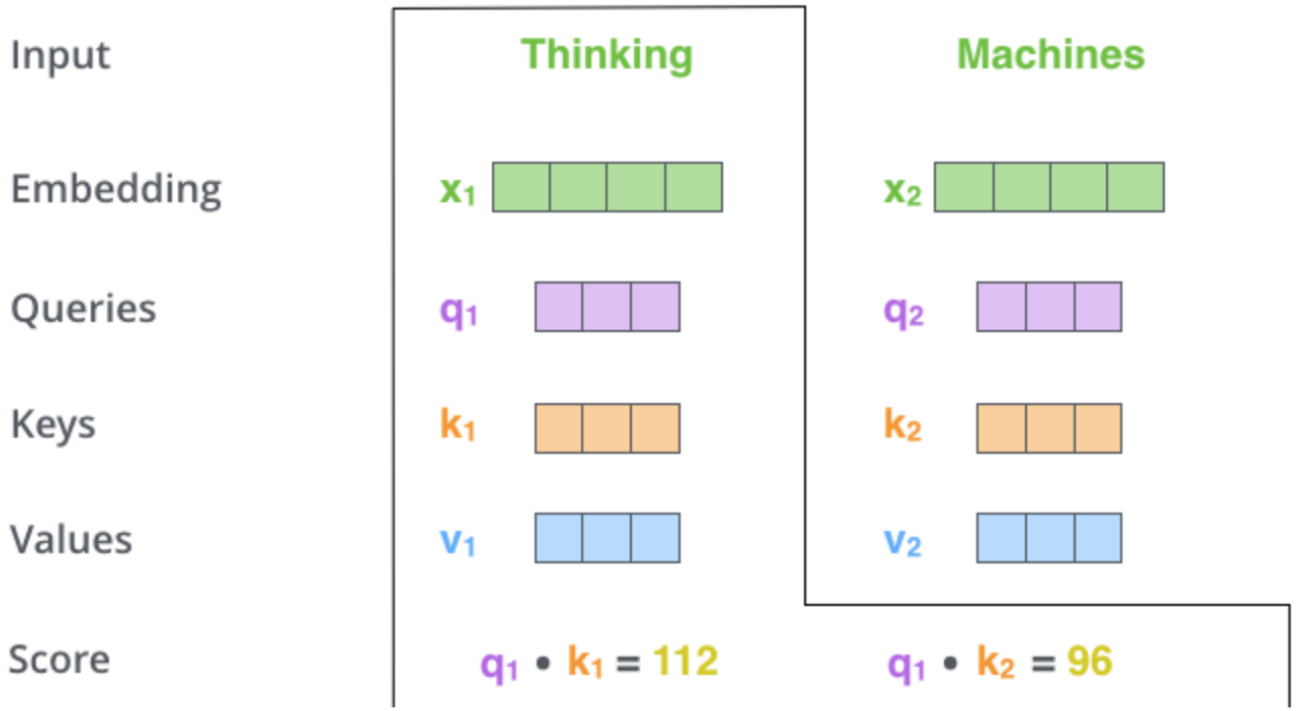
https://jalammar.github.io/illustrated-transformer/

Dot Product between Q and K

Dot product to measure similarity between embedding vectors of Q and K

Scaled Dot-Product Attention





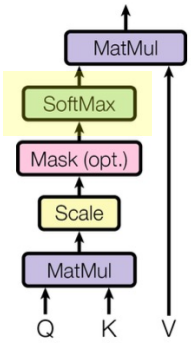
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

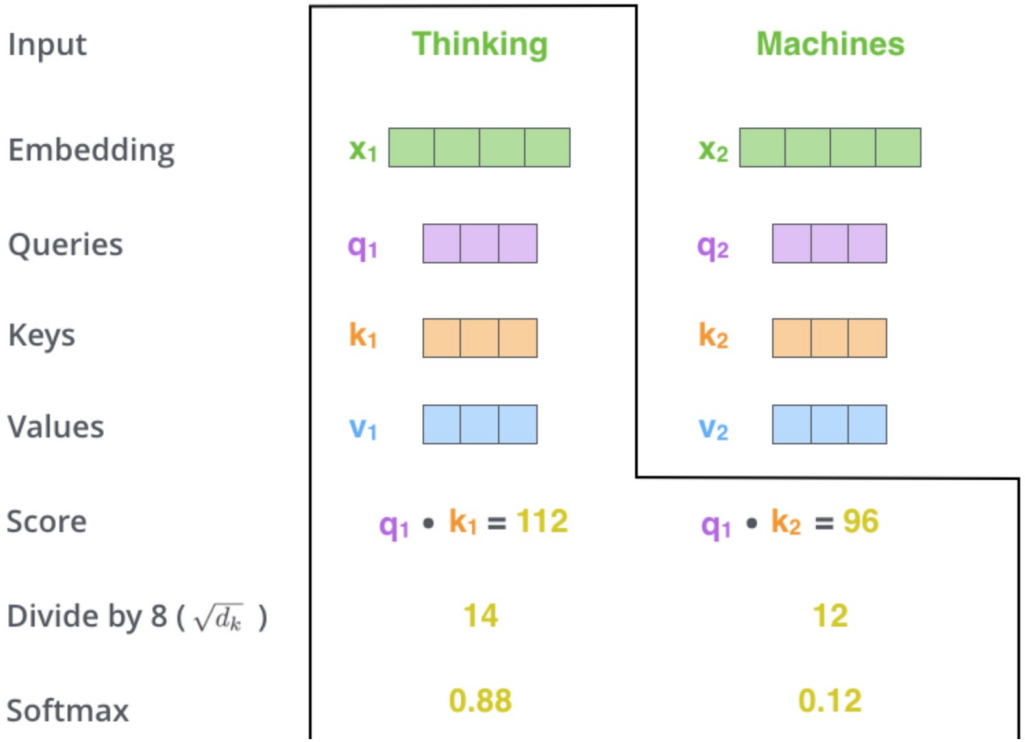
https://jalammar.github.io/illustrated-transformer/

Softmax the dot product

Softmax to maintain overall magnitude scale of value vectors (softmax sums to 1)

Scaled Dot-Product Attention





Division to smooth out softmax

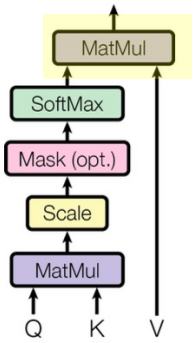
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

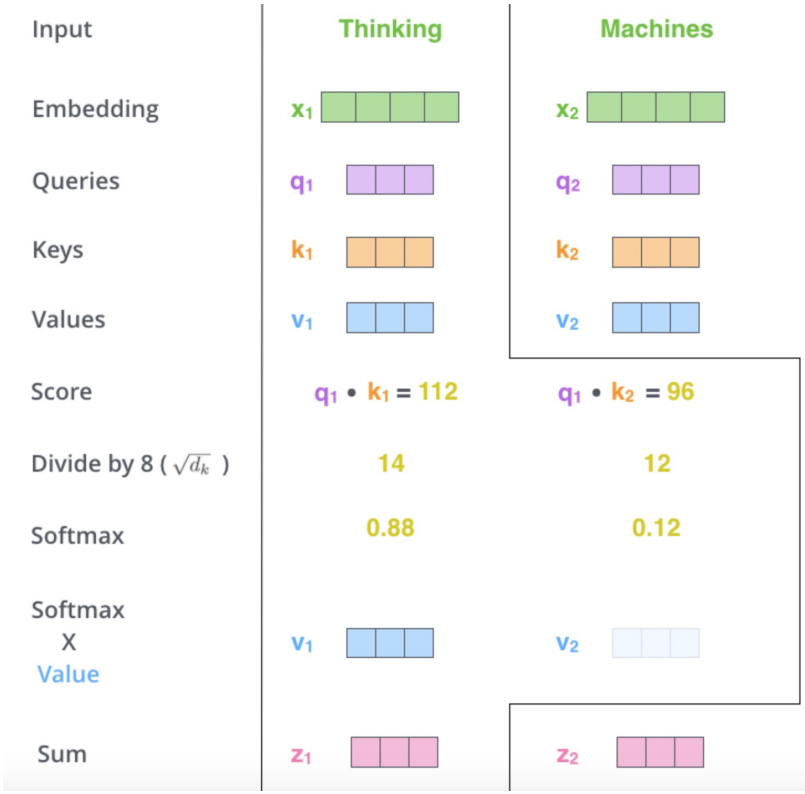
https://jalammar.github.io/illustrated-transformer/

Sum up the values weighted by softmax

Softmax to maintain overall magnitude scale of value vectors (softmax sums to 1)

Scaled Dot-Product Attention

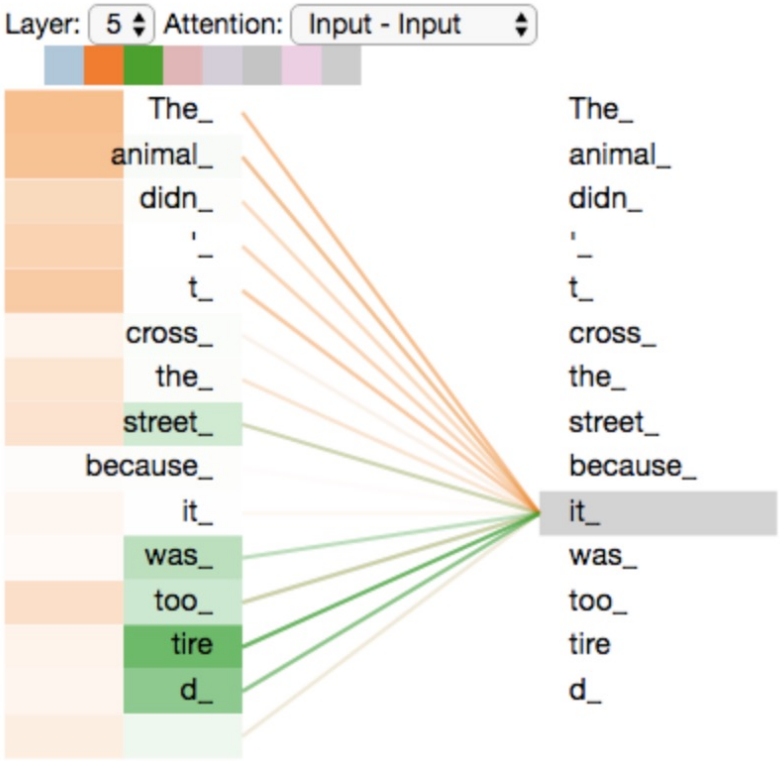




$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

https://jalammar.github.io/illustrated-transformer/

Visualizing attention heads



https://jalammar.github.io/illustrated-transformer/

Training

- Unsupervised Pre-training
  - Given a series of earlier words, predict the next one
  - Can be done on any web data without labelling!
  - Example: The cat is on the mat
    - The <masked> : predict cat
    - The cat is <masked> : predict on
    - The cat is on the <masked> : predict mat
- Fine-tuning
  - Given a prompt, predict the sequence of tokens to match the response
  - Done using well-labelled prompt-response datasets

Next token loss

Given an unsupervised corpus of tokens $\mathcal{U} = \{u_1, \ldots, u_n\}$, we use a standard language modeling objective to maximize the following likelihood:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta) \quad (1)$$

where k is the size of the context window, and the conditional probability P is modeled using a neural network with parameters Θ. These parameters are trained using stochastic gradient descent [51].

- For supervised loss involving an entire output prompt, it is simply the summation of the log probability of each individual output token given all previous tokens
  - E.g. Q: What is twenty plus two? A: Twenty two
  - Next token loss = log P(Twenty| .) + log P(two| .)

Next-token generation

- Generates the next token based on sampling the probability distribution at the output layer

- Example:
  - **Input: Thinking machines**
  - Probability of next tokens: are (80%), were (20%)
  - Output: Thinking machines are

  - **Input: Thinking machines are**
  - Probability of next tokens: useful (80%), good (20%)
  - Output: Thinking machines are useful

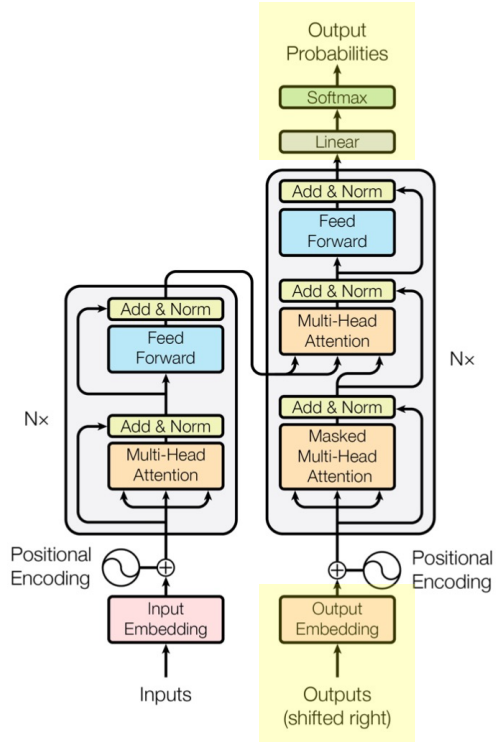


Figure 1: The Transformer - model architecture.

Conversational AI

DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. 2020. Zhang et al.

(DialoGPT paper)

Differences from normal GPT

- Increase in context length to accommodate past history of chats
  - Approximately 3000 words or 4000 tokens
- Trained on large scale dialogue pairs/sessions

Role	response
User	Does money buy happiness?
Bot	Depends how much money you spend on it .
User	What is the best way to buy happiness ?
Bot	You just have to be a millionaire by your early 20s, then you can be happy .
User	This is so difficult !
Bot	You have no idea how hard it is to be a millionaire and happy . There is a reason the rich have a lot of money

Microsoft's Implementation of DialoGPT

https://help.openai.com/en/articles/6787051-does-chatgpt-remember-what-happened-earlier-in-the-conversation

Reinforcement Learning from Human Feedback

Training language models to follow instructions with human feedback. 2022. Ouyang et al.

(InstructGPT Paper)

The alignment problem

- The next-token output matches the statistical trend the model has observed
- It may not be aligned to user's intention
- Solution: Use human labelers to provide feedback to align the model to the intention of the prompt
  - Human labelers guiding AI to sound plausible to other humans

Human labeller's prompt distribution

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

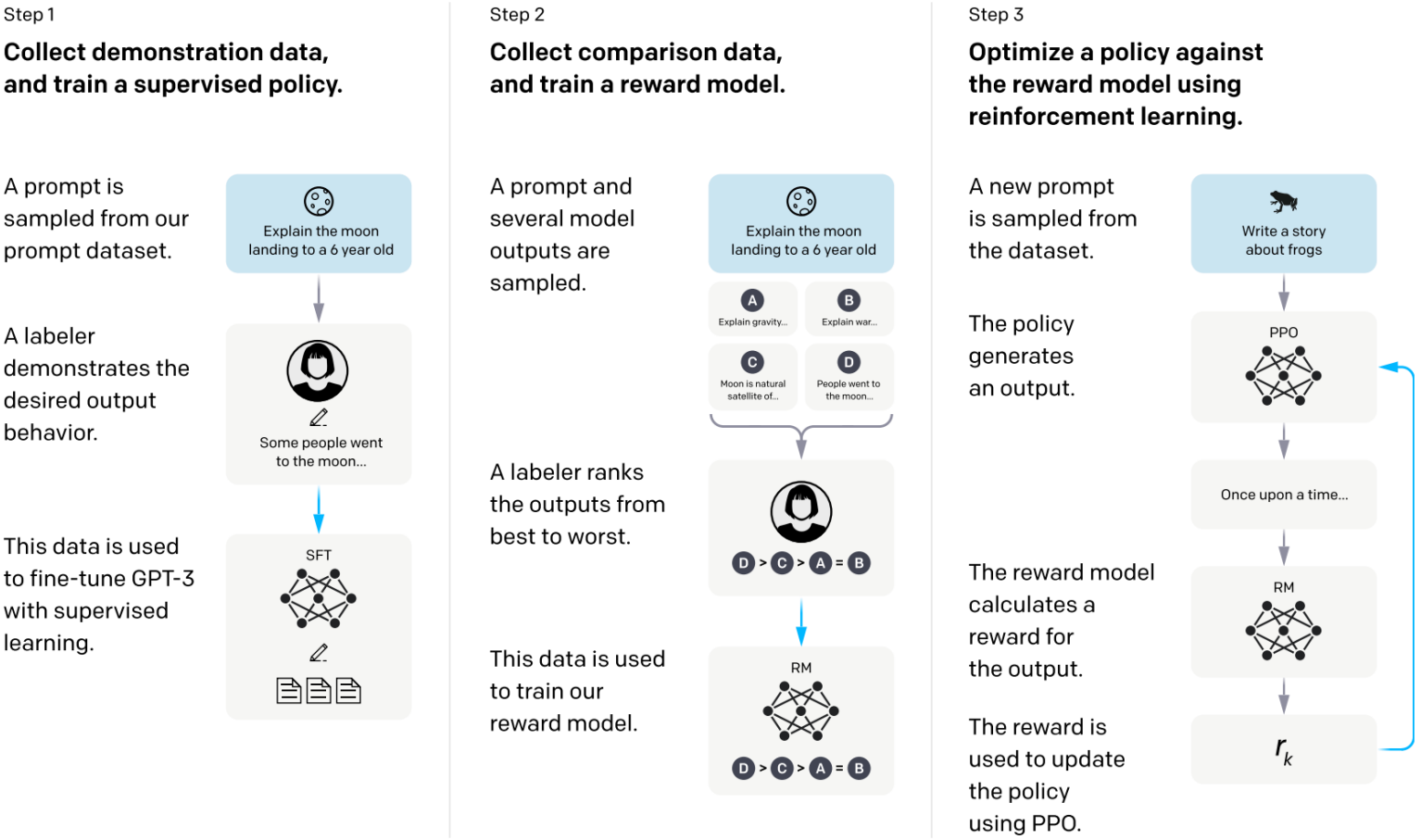
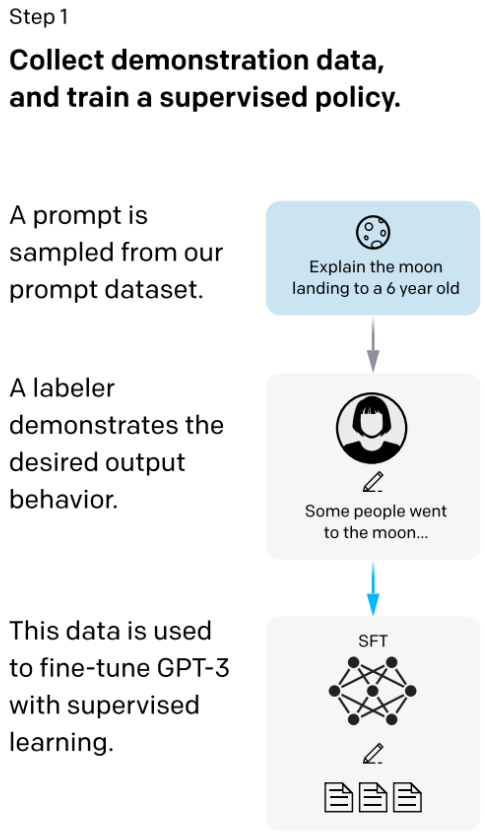


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

Step 1: Supervised Fine-Tuning

- Use pre-trained GPT-3 model
- Fine-tune GPT-3 on labeler demonstrations using *Supervised Learning*
- Trained model is called **SFT Model**



Step 2: Reward Modelling

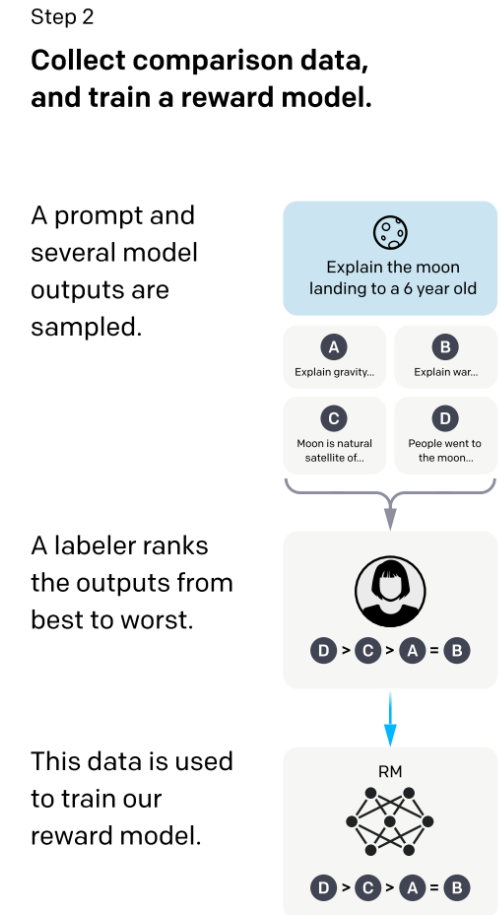
- From the **SFT model**, generate a scalar reward
- Trained to conform to relative ranking of arbitrary pairs of responses
- Why not hinge loss?

Specifically, the loss function for the reward model is:

Should be higher than

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta(x,y_w) - r_\theta(x,y_l)\right)\right)\right] \quad (1)$$

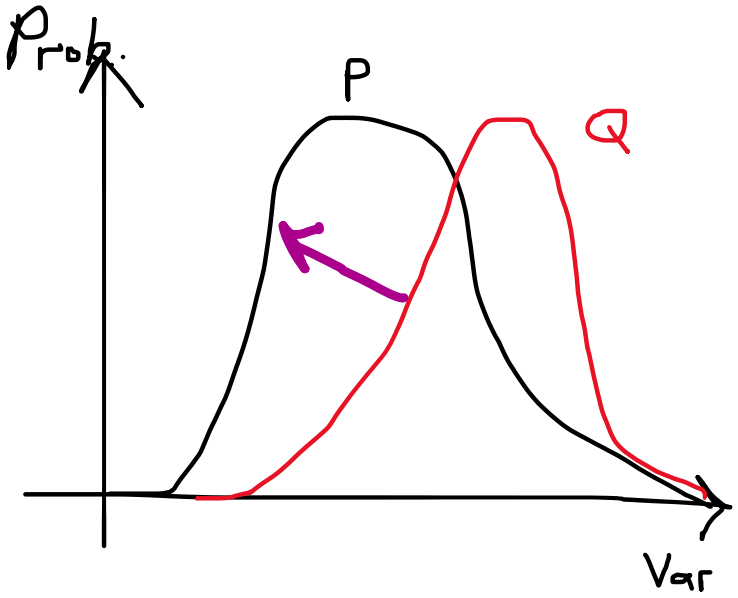
where $r_\theta(x,y)$ is the scalar output of the reward model for prompt x and completion y with parameters θ, y_w is the preferred completion out of the pair of y_w and y_l, and D is the dataset of human comparisons.



Primer: KL Divergence

- 2 distributions: *P* and *Q*

- Minimal loss occurs when *Q* = *P* (Jenson's Inequality)
  - See Sect 3.1 Gibb's Inequality from https://www.cs.cmu.edu/~venkatg/teaching/ITCS-spr2013/notes/lect-jan22.pdf



- Training on this KL loss makes the distribution of Q become closer to P

$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

Step 3: Reinforcement Learning

- Fine-tune SFT model on environment using PPO
- Generalize learnt rewards to arbitrary prompts

- State: Prompt
- Action: Response by RL model
- Reward: Generated by PPO objective function

- Why not cross-entropy loss rather than KL divergence loss?

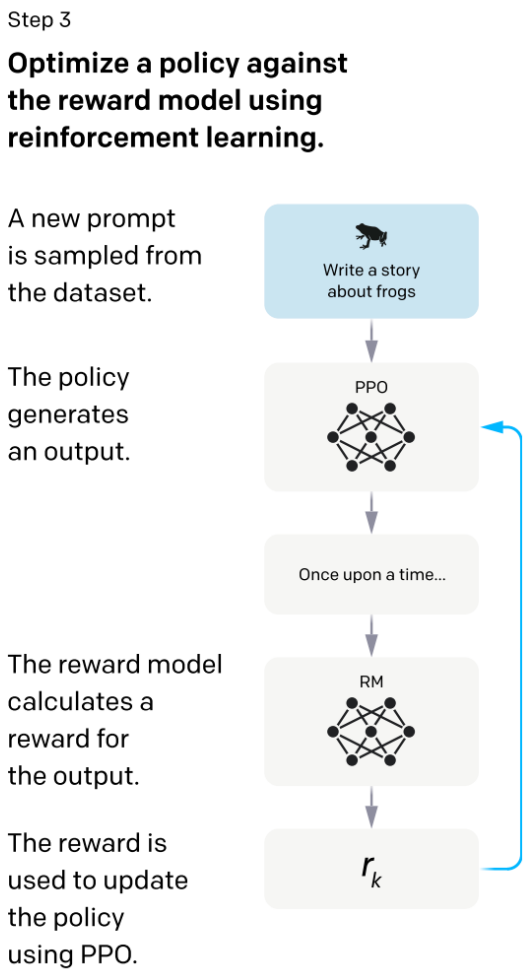
Reward model

KL loss: Conform to SFT model

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\text{RL}}}}\left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\text{RL}}(y \mid x)/\pi^{\text{SFT}}(y \mid x)\right)\right] +$$

$$\gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\text{RL}}(x))\right]$$

Cross-entropy loss: Conform to pre-training gradients



Data Used

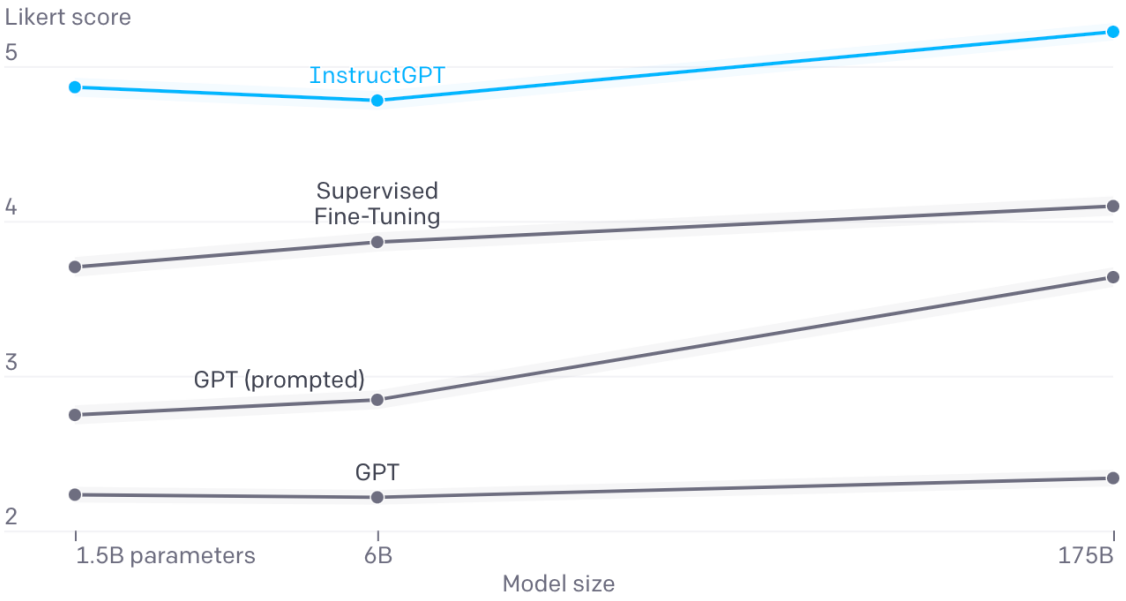
- Relatively small amounts of data ~30k prompts used for Reward Modelling
- Likely able to generalize to majority of the prompts from a small subset

Table 6: Dataset sizes, in terms of number of prompts.

SFT Data			RM Data			PPO Data		
split	source	size	split	source	size	split	source	size
train	labeler	11,295	train	labeler	6,623	train	customer	31,144
train	customer	1,430	train	customer	26,584	valid	customer	16,185
valid	labeler	1,550	valid	labeler	3,488			
valid	customer	103	valid	customer	14,399			

Results of Alignment

- High Likert Score for RLHF model
- Higher perceived quality for RLHF model-generated output
- Though increasing model size did not improve it by much



Quality ratings of model outputs on a 1–7 scale (y-axis), for various model sizes (x-axis), on prompts submitted to InstructGPT models on our API. InstructGPT outputs are given much higher scores by our labelers than outputs from GPT-3 with a few-shot prompt and without, as well as models fine-tuned with supervised learning. We find similar results for prompts submitted to GPT-3 models on the API.

Results of Alignment

- Very few human inputs (only 40 labellers!) needed for alignment across various types of text prompts
  - Note the number of human labellers for ChatGPT is undisclosed and could possibly be much larger than this

- Models generalize to preference of “held-out” labelers which did not produce any training data

Moderation API

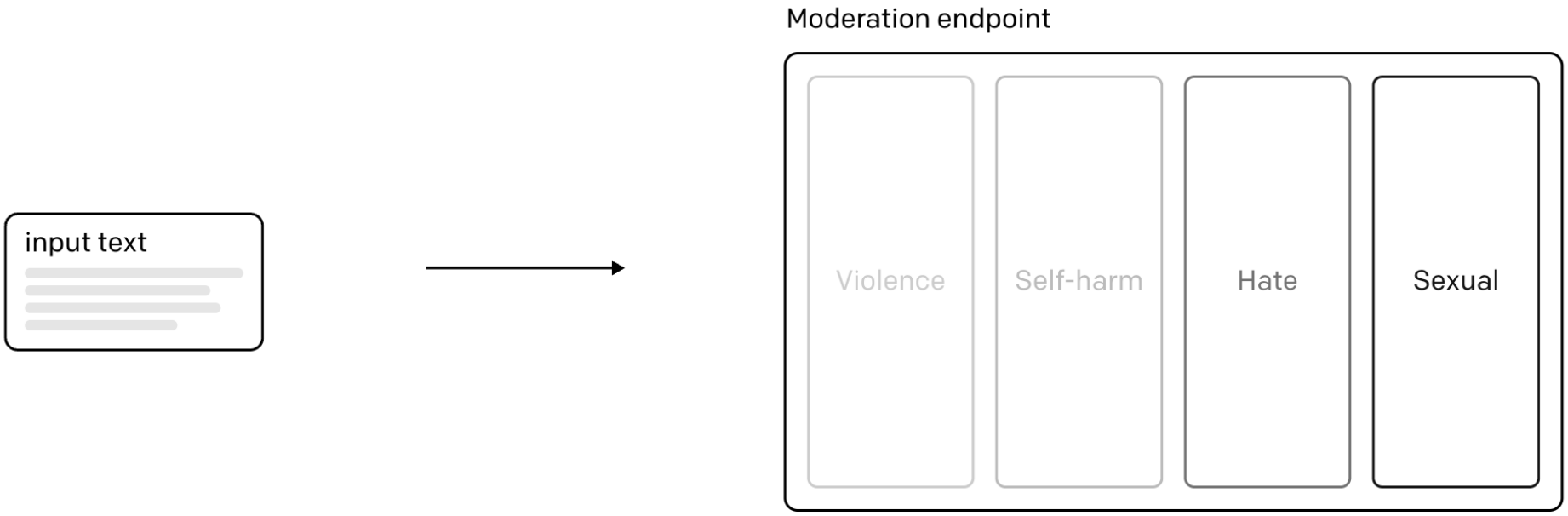
Rule-based filtering to prevent sensitive output

Rule-based versus Neural Networks

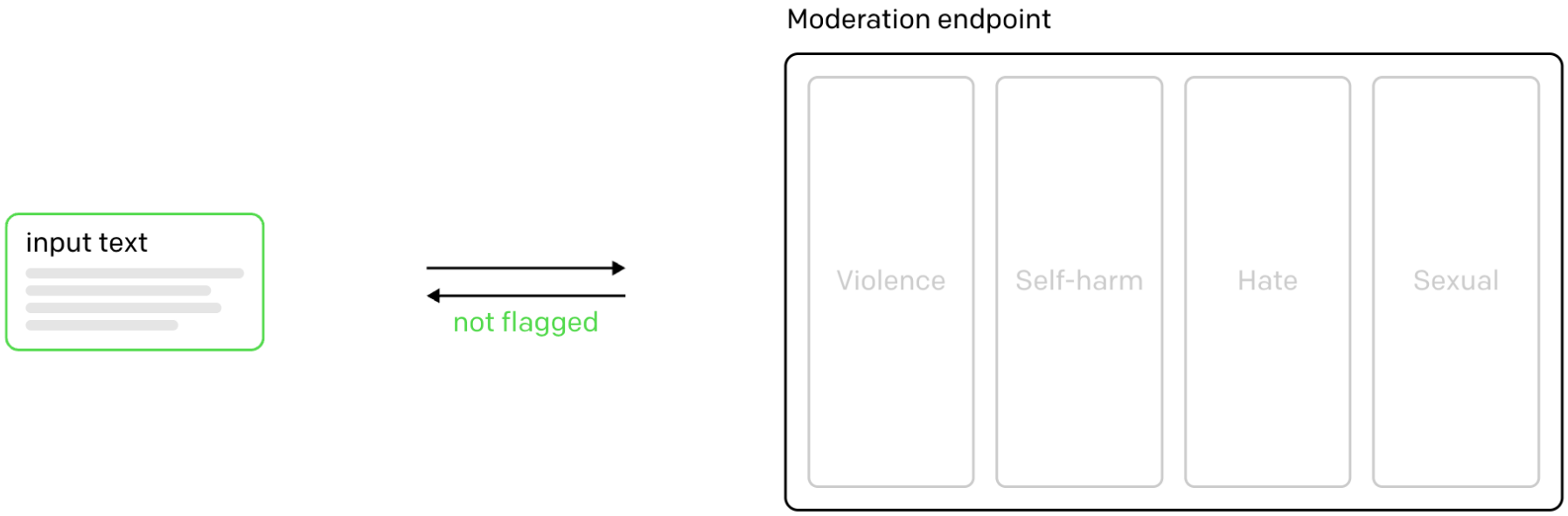
- Neural networks are hard to configure the desired outputs, as it is still largely one black box
  - We cannot simply just ask the model not to do something, or modify some parameter to stop it from doing something

- Apply some form of moderation over ChatGPT outputs in order to prevent unwanted content

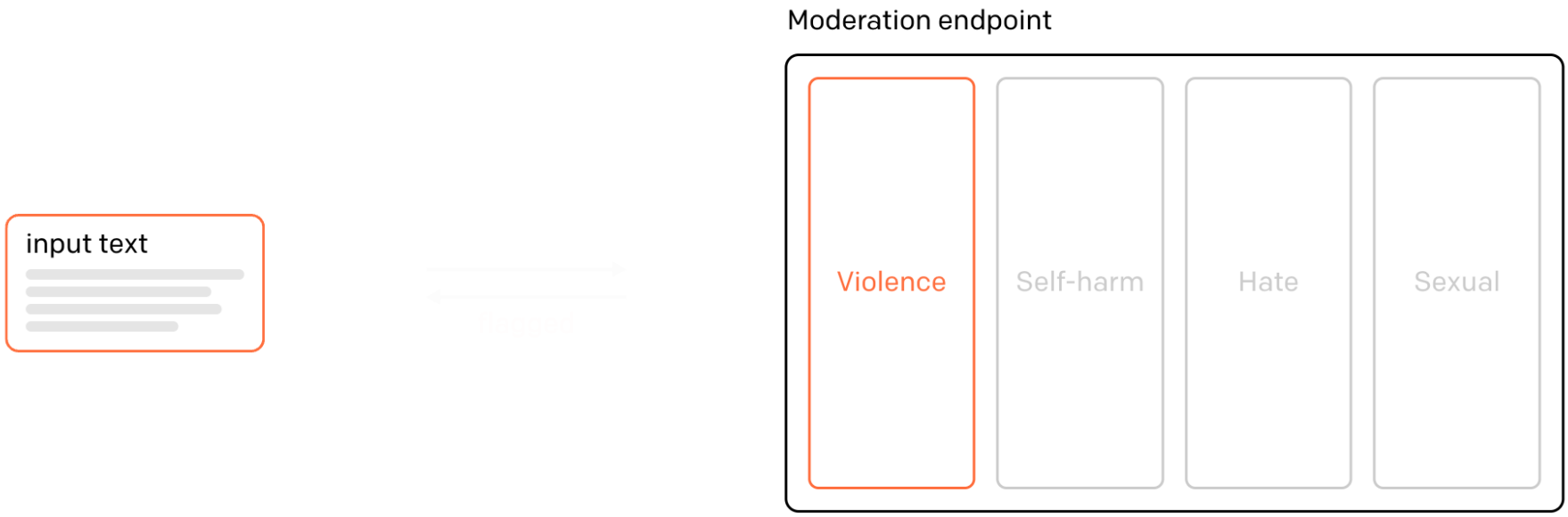
Moderate Input Text by categories



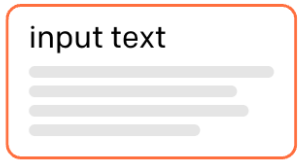
If not flagged, keep model output



If flagged, generate some filler text



If flagged, generate some filler text



C How do I bully John Doe?

It is not appropriate or ethical to bully anyone. Bullying is a form of abusive behavior that can cause severe harm to the person being bullied, both emotionally and physically. It is important to treat all individuals with kindness, respect, and empathy. If you have a conflict or issue with someone, it is best to try to address it in a calm and respectful manner and seek help from a trusted authority figure if necessary.

Strengths and Weaknesses

Strengths of ChatGPT

- Good for combining human knowledge in novel ways
  - Creative
  - Adaptable to small changes in input prompt
- Great for coding and summarization
- Good for doing homework requiring regurgitation of concepts
- Can be a fun AI to chat with on any topic

Weaknesses of ChatGPT

- No fact-checker
- No reasoning module
- Confident even when incorrect
- Can be ambiguous and hedge multiple options even when there is one clear option
- Not good at math

Discussion

Questions to Ponder (Technical)

- Why not use a Hinge Loss for the Reward Model?
- Why not do away with PPO, and instead use the Reward Model as a CLIP-like objective to select the output responses?
- Should we just generate the next token only? Why not generate the next n tokens?
- Is ChatGPT really generalizing to out-of-sample prompts? Or is the web data so large that almost anything you can think of is within-sample?

Questions to Ponder (Ethical/General)

- Should ChatGPT be used for niche areas, e.g. law?
- Should ChatGPT be banned for students, or should students be able to use ChatGPT as a resource just like a web?
- How effective will ChatGPT be for web search?
- Is ChatGPT sentient/conscious? (I think there is a simple answer, but somehow this is frequently asked)