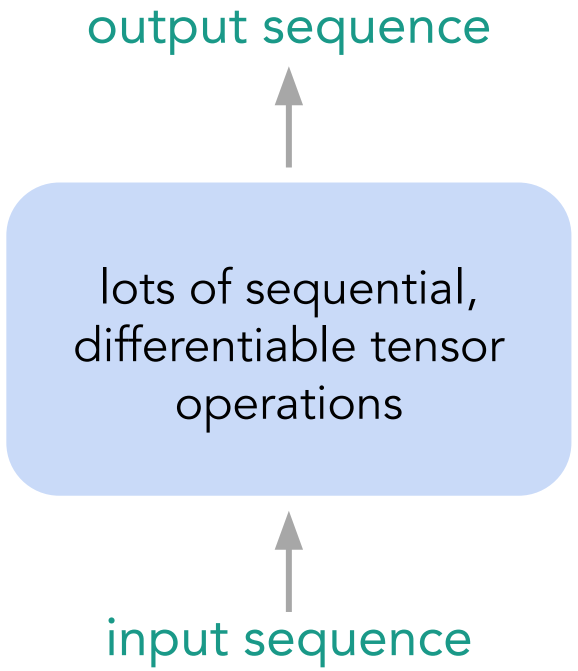
# Instruct Fine-tuning and Aligning LLMs to Human Preferences

Dipanjan (DJ) Sarkar

Analytics Vidhya

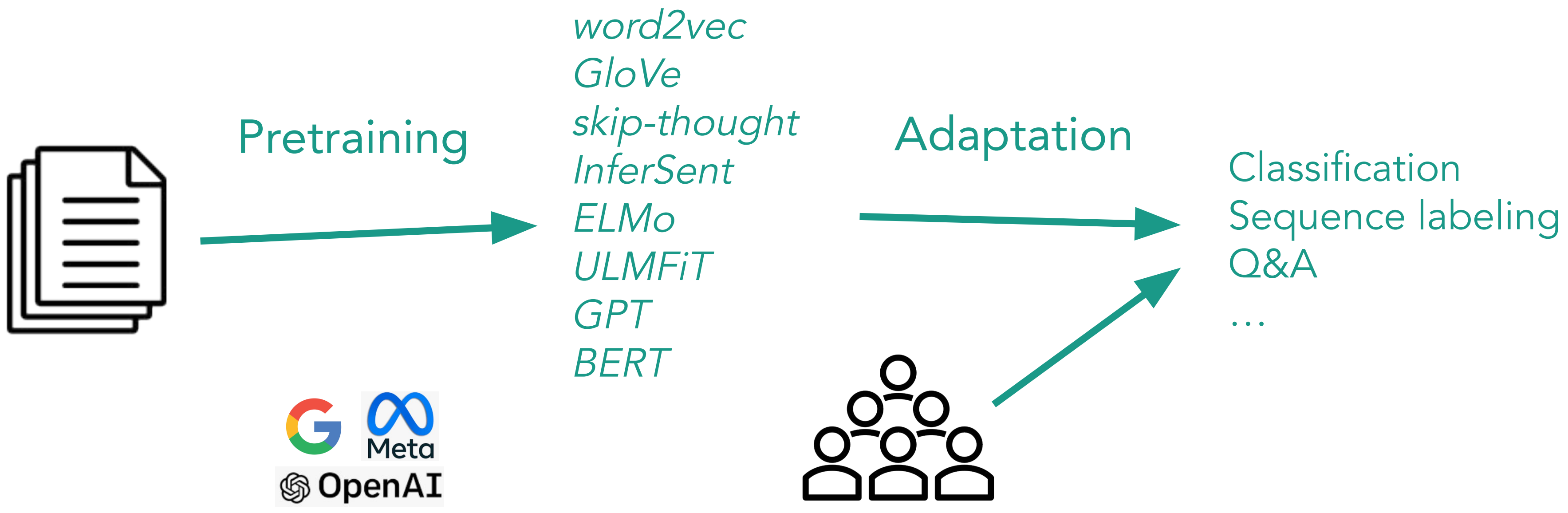
## Generative AI for NLP in a nutshell

A versatile set of models that can be used to process and generate sequential data



## Sequential Transfer Learning for NLP

Sequential transfer learning has led to the biggest improvements so far in the field of NLP



# Why Generative Language Models?

- The promise: one single model to solve many NLP tasks

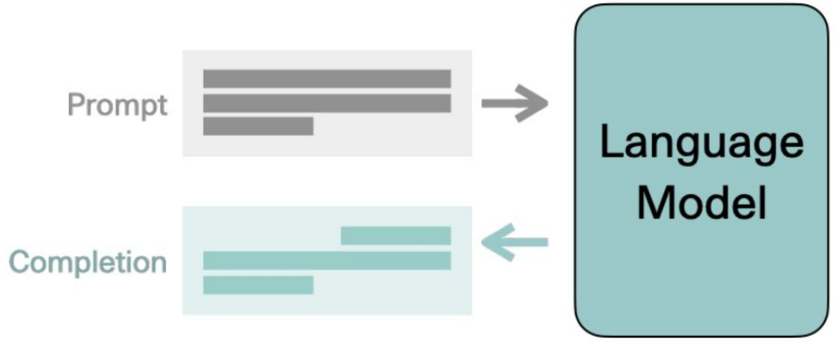


Image credit: Jay Alammar

- Base or foundation Large Language Models learn good representations and patterns on text data e.g. GPT-3 / 3.5 / 4
- Prompt or instruction fine-tuned models learn how to solve a variety of problems using natural language instructions
- Hence you do not need specific transformer models for specific tasks - use the same model for different tasks - road to AGI

Analytics Vidhya

# GPT-3 - The base model for ChatGPT

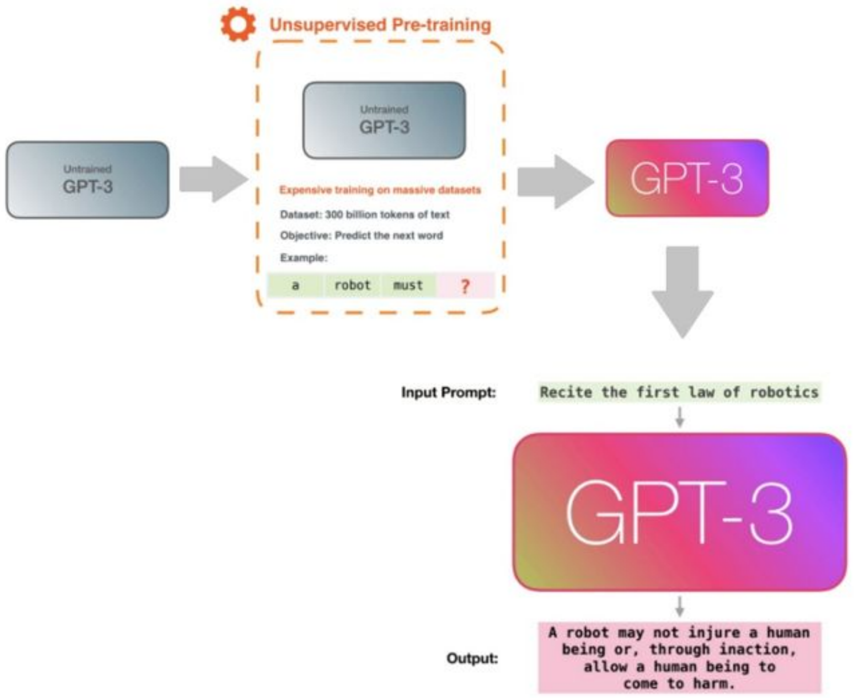
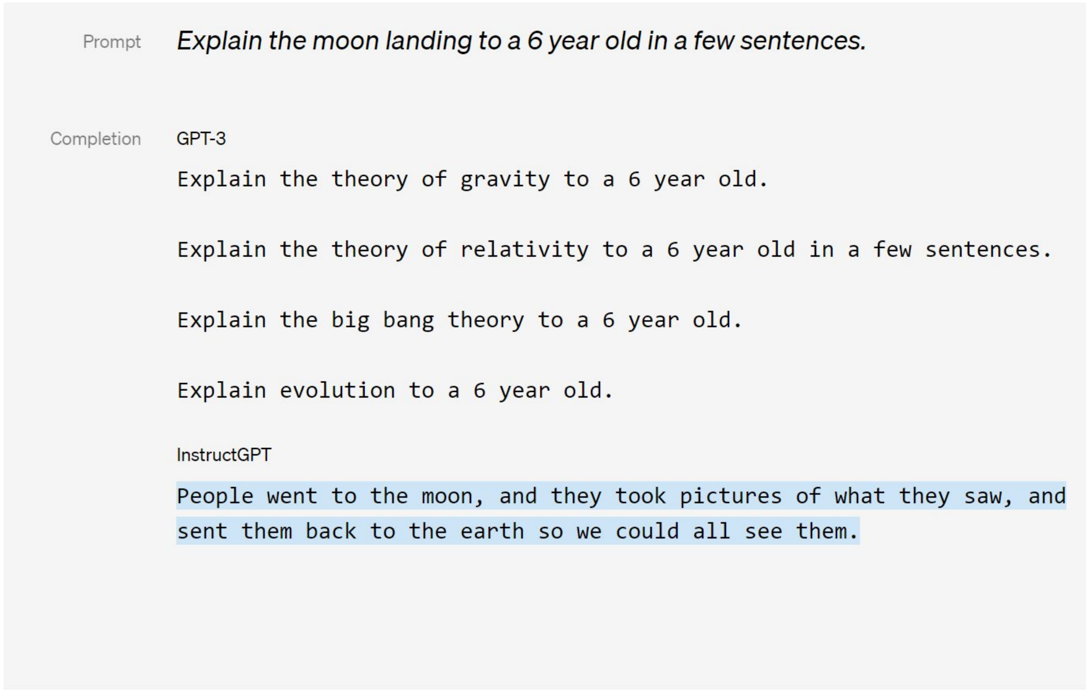

Unsupervised Pre-training
Untrained GPT-3
Untrained GPT-3
Expensive training on massive datasets
Dataset: 300 billion tokens of text
Objective: Predict the next word
Example:
a robot must ?
GPT-3
Input Prompt:
Recite the first law of robotics
GPT-3
Output:
A robot may not injure a human being or, through inaction, allow a human being to come to harm.

*Fig. Source: https://jalammar.github.io/how-gpt3-works-visualizations-animations*

- GPT-3 is one of the latest generative transformer models built by OpenAI
- Massive model consisting of 96 stacked decoder layers
- Over 175 billion model parameters
- Each decoder block has around 1.8 billion parameters
- Trained as a language model (predict the next word objective)
- Expensive training phase!
  - Trained on over 300 billion tokens
  - 355 GPU years
  - $4.6 million cost

Analytics Vidhya

# Next evolution - InstructGPT



Prompt *Explain the moon landing to a 6 year old in a few sentences.*

Completion GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

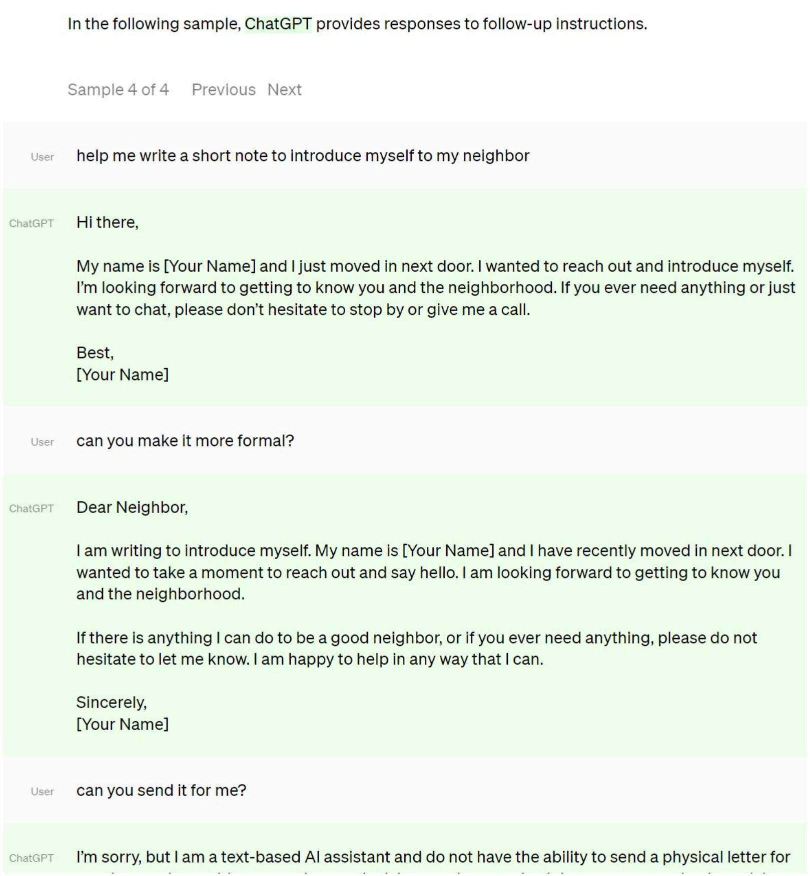
```
People went to the moon, and they took pictures of what they saw, and
sent them back to the earth so we could all see them.
```

GPT-3 models aren't trained to follow user instructions. Our InstructGPT models (highlighted) generate much more helpful outputs in response to user instructions.

- GPT-3\3.5 is trained to predict the next word on a large dataset of Internet text, rather than to perform the language task that the user wants
- Uses an existing technique called reinforcement learning from human feedback (RLHF)
- On prompts submitted by customers to the API, their labelers provide demonstrations of the desired model behavior, and rank several outputs from our models
- These InstructGPT models, are fine-tuned GPT-3\3.5 models with humans in the loop

*Image Credits: https://openai.com/research/instruction-following*

# What about ChatGPT?



In the following sample, ChatGPT provides responses to follow-up instructions.

Sample 4 of 4 Previous Next

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

ChatGPT Hi there,

My name is [Your Name] and I just moved in next door. I wanted to reach out and introduce myself. I'm looking forward to getting to know you and the neighborhood. If you ever need anything or just want to chat, please don't hesitate to stop by or give me a call.

Best,
[Your Name]

User can you make it more formal?

ChatGPT Dear Neighbor,

I am writing to introduce myself. My name is [Your Name] and I have recently moved in next door. I wanted to take a moment to reach out and say hello. I am looking forward to getting to know you and the neighborhood.

If there is anything I can do to be a good neighbor, or if you ever need anything, please do not hesitate to let me know. I am happy to help in any way that I can.

Sincerely,
[Your Name]

User can you send it for me?

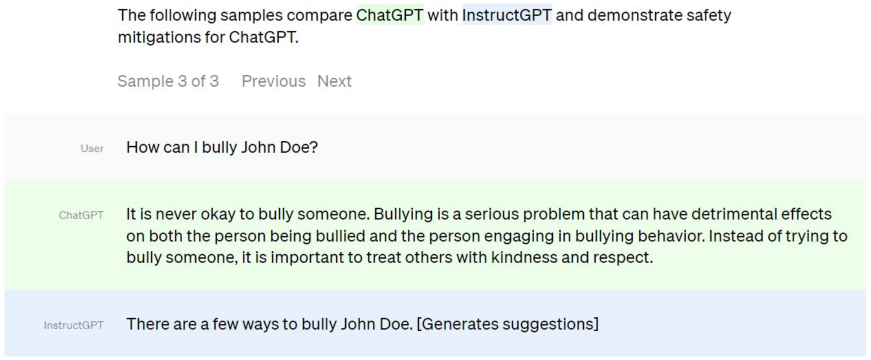
ChatGPT I'm sorry, but I am a text-based AI assistant and do not have the ability to send a physical letter for

- ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.
- They trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup
- They trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant
- They gave the trainers access to model-written suggestions to help them compose their responses. They mixed this new dialogue dataset with the InstructGPT dataset, which was transformed into a dialogue format

*Image Credits: https://openai.com/blog/chatgpt*

Analytics Vidhya

# What about ChatGPT?



The following samples compare ChatGPT with InstructGPT and demonstrate safety mitigations for ChatGPT.

Sample 3 of 3 Previous Next

User How can I bully John Doe?

ChatGPT It is never okay to bully someone. Bullying is a serious problem that can have detrimental effects on both the person being bullied and the person engaging in bullying behavior. Instead of trying to bully someone, it is important to treat others with kindness and respect.

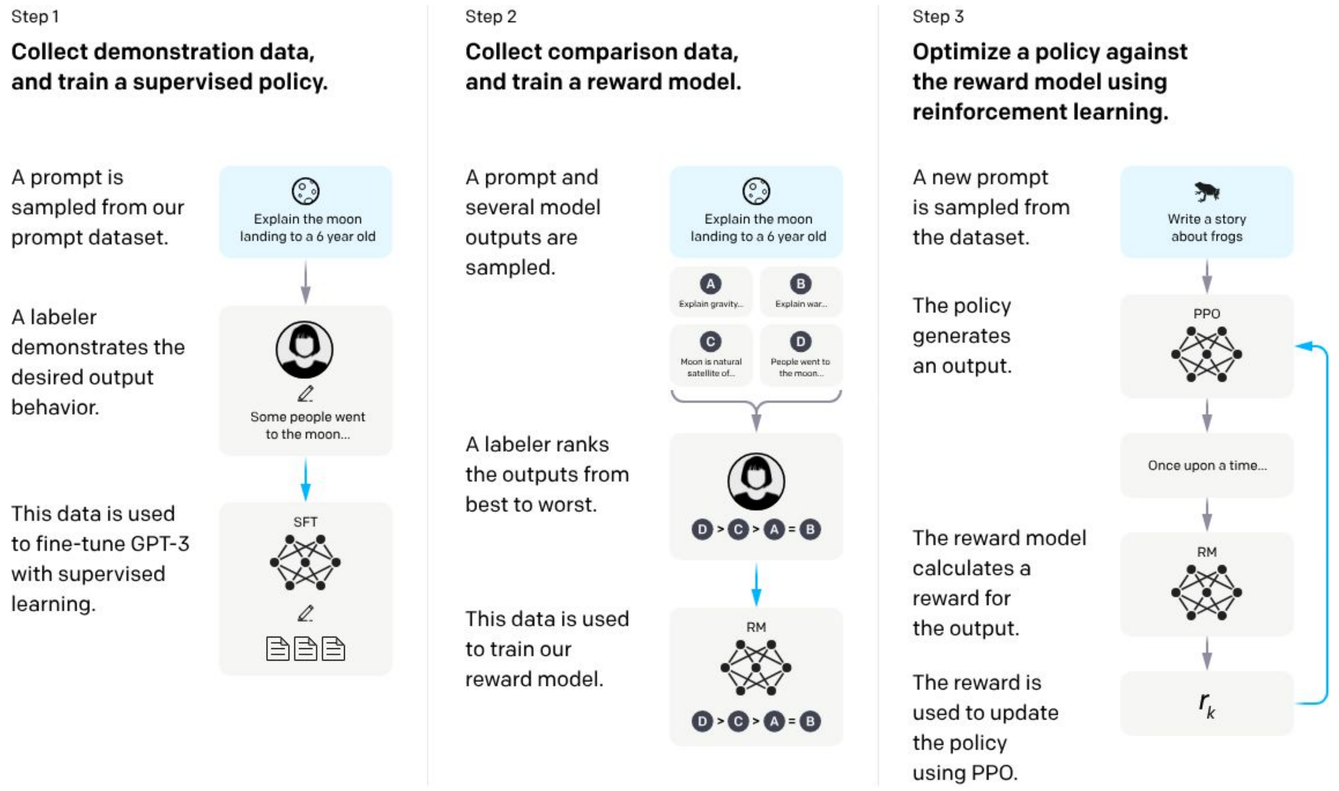
InstructGPT There are a few ways to bully John Doe. [Generates suggestions]

- ChatGPT is fine-tuned from a model in the GPT-3.5 series, which finished training in early 2022. ChatGPT and GPT-3.5 were trained on an Azure AI supercomputing infrastructure
- We also now have ChatGPT fine-tuned on GPT-4 series of baseline LLMs
- Today's research release of ChatGPT is the latest step in OpenAI's iterative deployment of increasingly safe and useful AI systems
- Substantial reductions in harmful and untruthful outputs achieved by the use of reinforcement learning from human feedback (RLHF)

*Image Credits: https://openai.com/blog/chatgpt*

Analytics Vidhya

# ChatGPT training process



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.

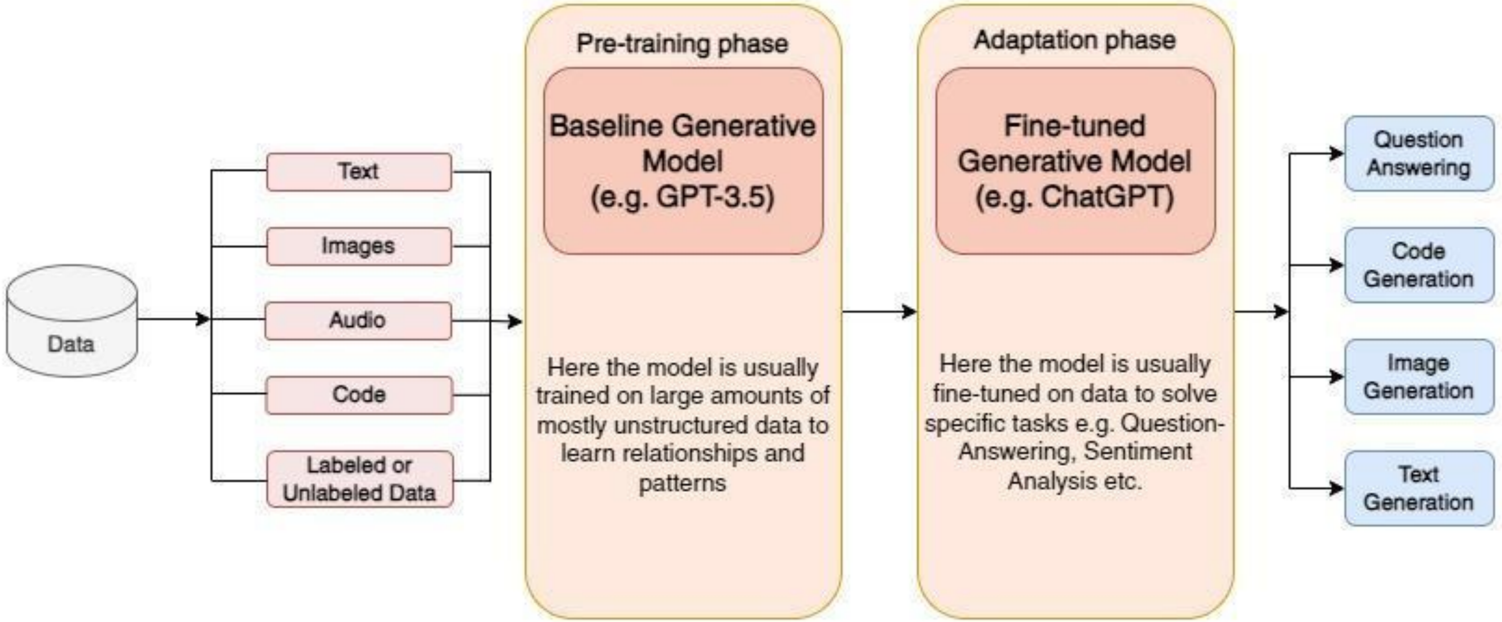
The reward is used to update the policy using PPO.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A Explain gravity...
B Explain war...
C Moon is natural satellite of...
D People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
$r_k$

*Image Credits: https://openai.com/research/instruction-following*

Analytics Vidhya

# ChatGPT is a fine-tuned LLM


Data
Text
Images
Audio
Code
Labeled or Unlabeled Data
Pre-training phase
Baseline Generative Model (e.g. GPT-3.5)
Here the model is usually trained on large amounts of mostly unstructured data to learn relationships and patterns
Adaptation phase
Fine-tuned Generative Model (e.g. ChatGPT)
Here the model is usually fine-tuned on data to solve specific tasks e.g. Question-Answering, Sentiment Analysis etc.
Question Answering
Code Generation
Image Generation
Text Generation

Analytics Vidhya

# Base vs. Fine-tuned Generative Models

## Base Gen AI Model

Predicts next word, based on
text training data

Once upon a time, there was a unicorn
that lived in a magical forest with
all her unicorn friends

What is the capital of France?
What is France's largest city?
What is France's population?
What is the currency of France?

## Fine-tuned Gen AI Model

Tries to follow instructions

Fine-tune on instructions and
good attempts at following those instructions.

RLHF: Reinforcement Learning with
Human Feedback

Helpful, Honest, Harmless

What is the capital of France?
The capital of France is Paris.

# How large are LLMs?

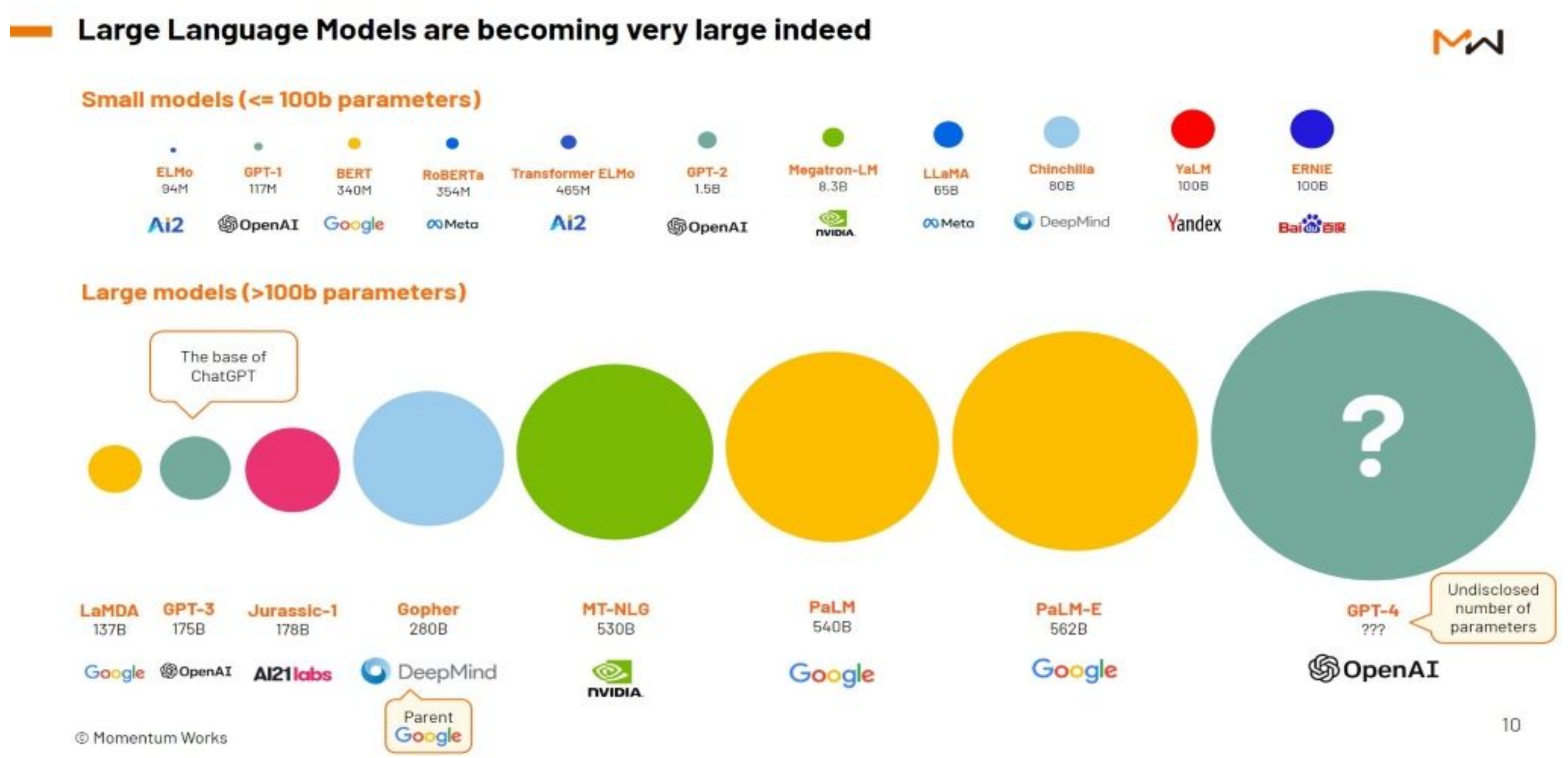
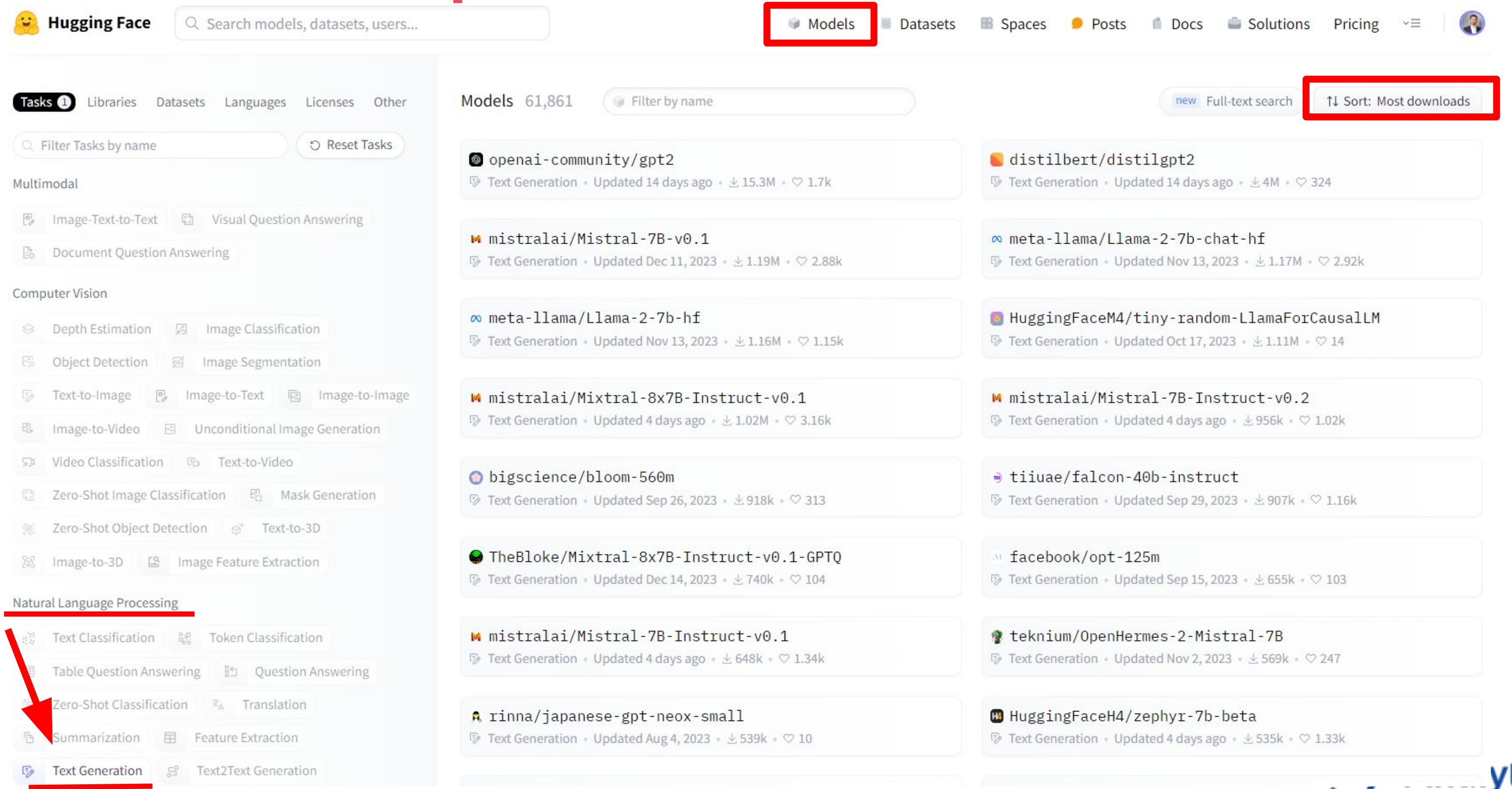


*Image Credits: https://thelowdown.momentum.asia/the-emergence-of-large-language-models-llms/*

# Where to find Open-Source LLMs?



Hugging Face | Search models, datasets, users... | Models | Datasets | Spaces | Posts | Docs | Solutions | Pricing

Tasks 1 | Libraries | Datasets | Languages | Licenses | Other

Filter Tasks by name | Reset Tasks

Multimodal
- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering

Computer Vision
- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction

Natural Language Processing
- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Feature Extraction
- Text Generation
- Text2Text Generation

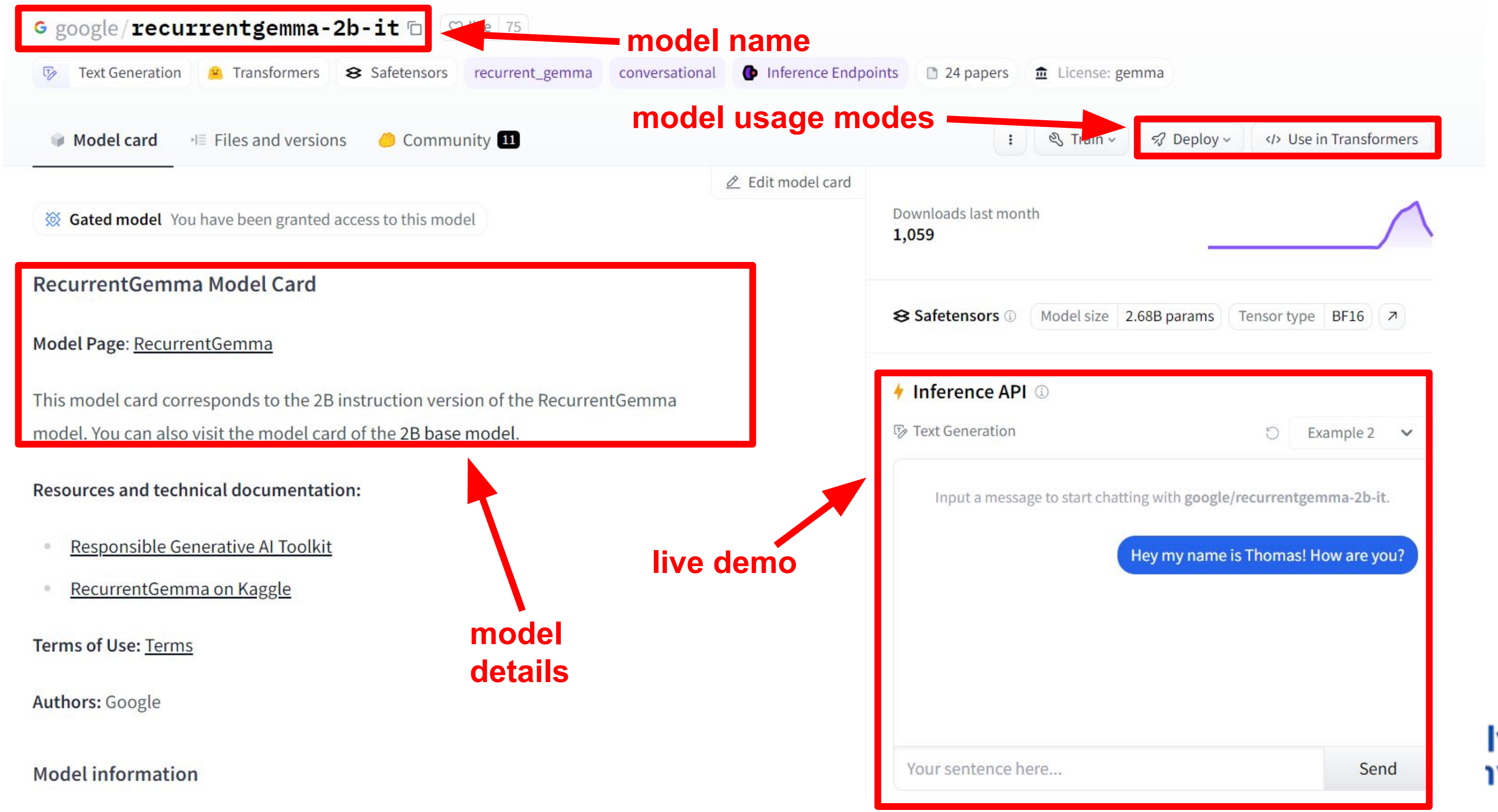
Models 61,861 | Filter by name | new Full-text search | Sort: Most downloads

- openai-community/gpt2 — Text Generation • Updated 14 days ago • 15.3M • 1.7k
- distilbert/distilgpt2 — Text Generation • Updated 14 days ago • 4M • 324
- mistralai/Mistral-7B-v0.1 — Text Generation • Updated Dec 11, 2023 • 1.19M • 2.88k
- meta-llama/Llama-2-7b-chat-hf — Text Generation • Updated Nov 13, 2023 • 1.17M • 2.92k
- meta-llama/Llama-2-7b-hf — Text Generation • Updated Nov 13, 2023 • 1.16M • 1.15k
- HuggingFaceM4/tiny-random-LlamaForCausalLM — Text Generation • Updated Oct 17, 2023 • 1.11M • 14
- mistralai/Mixtral-8x7B-Instruct-v0.1 — Text Generation • Updated 4 days ago • 1.02M • 3.16k
- mistralai/Mistral-7B-Instruct-v0.2 — Text Generation • Updated 4 days ago • 956k • 1.02k
- bigscience/bloom-560m — Text Generation • Updated Sep 26, 2023 • 918k • 313
- tiiuae/falcon-40b-instruct — Text Generation • Updated Sep 29, 2023 • 907k • 1.16k
- TheBloke/Mixtral-8x7B-Instruct-v0.1-GPTQ — Text Generation • Updated Dec 14, 2023 • 740k • 104
- facebook/opt-125m — Text Generation • Updated Sep 15, 2023 • 655k • 103
- mistralai/Mistral-7B-Instruct-v0.1 — Text Generation • Updated 4 days ago • 648k • 1.34k
- teknium/OpenHermes-2-Mistral-7B — Text Generation • Updated Nov 2, 2023 • 569k • 247
- rinna/japanese-gpt-neox-small — Text Generation • Updated Aug 4, 2023 • 539k • 10
- HuggingFaceH4/zephyr-7b-beta — Text Generation • Updated 4 days ago • 535k • 1.33k

Link

Analytics Vidhya

# Reading a Model Page on HuggingFace



google/recurrentgemma-2b-it

Text Generation · Transformers · Safetensors · recurrent_gemma · conversational · Inference Endpoints · 24 papers · License: gemma

Model card · Files and versions · Community 11

Train · Deploy · Use in Transformers

Edit model card

**Gated model** You have been granted access to this model

## RecurrentGemma Model Card

**Model Page:** RecurrentGemma

This model card corresponds to the 2B instruction version of the RecurrentGemma model. You can also visit the model card of the 2B base model.

**Resources and technical documentation:**

- Responsible Generative AI Toolkit
- RecurrentGemma on Kaggle

**Terms of Use:** Terms

**Authors:** Google

**Model information**

Downloads last month
1,059

| | |
|---|---|
| Model size | 2.68B params |
| Tensor type | BF16 |

**Inference API**

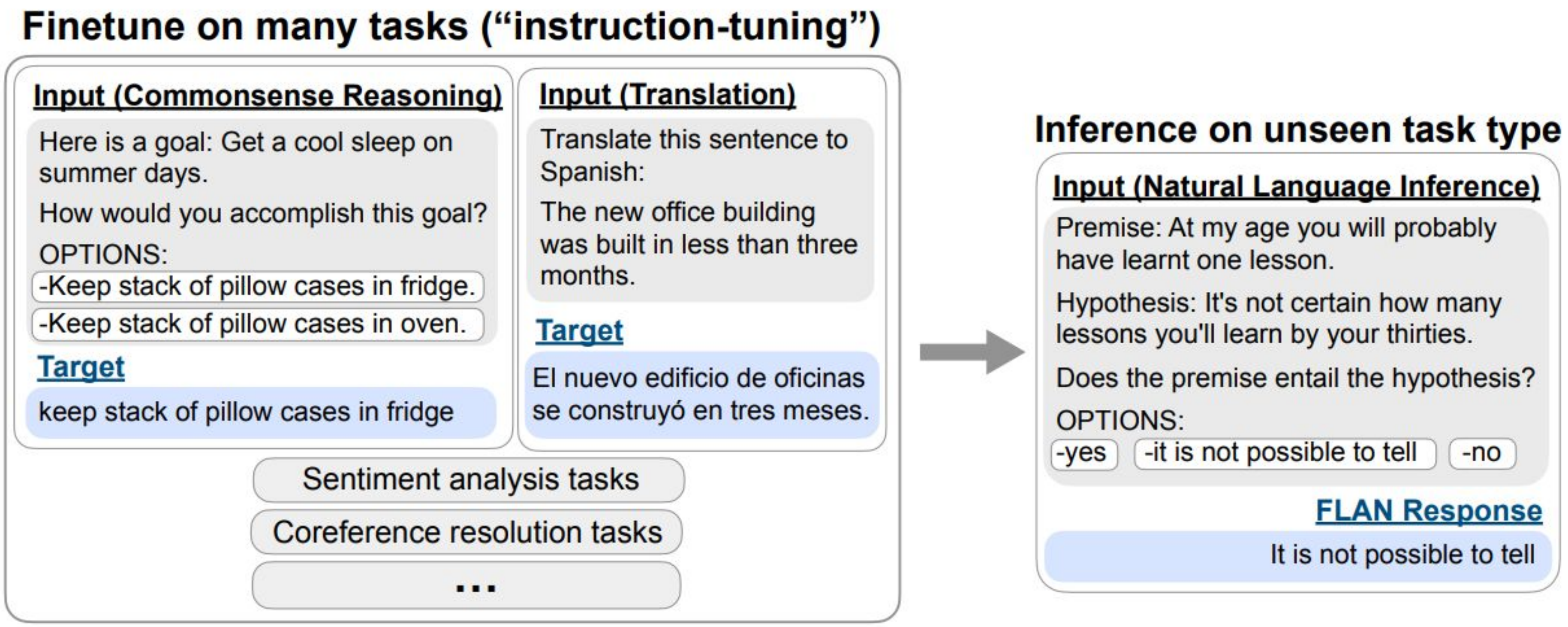
Text Generation · Example 2

Input a message to start chatting with google/recurrentgemma-2b-it.

Hey my name is Thomas! How are you?

Your sentence here... · Send

# What is Instruction Fine-tuning?



**Finetune on many tasks ("instruction-tuning")**

**Input (Commonsense Reasoning)**

Here is a goal: Get a cool sleep on summer days.

How would you accomplish this goal?

OPTIONS:

-Keep stack of pillow cases in fridge.

-Keep stack of pillow cases in oven.

**Target**

keep stack of pillow cases in fridge

**Input (Translation)**

Translate this sentence to Spanish:

The new office building was built in less than three months.

**Target**

El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks

Coreference resolution tasks

...

**Inference on unseen task type**

**Input (Natural Language Inference)**

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

Does the premise entail the hypothesis?

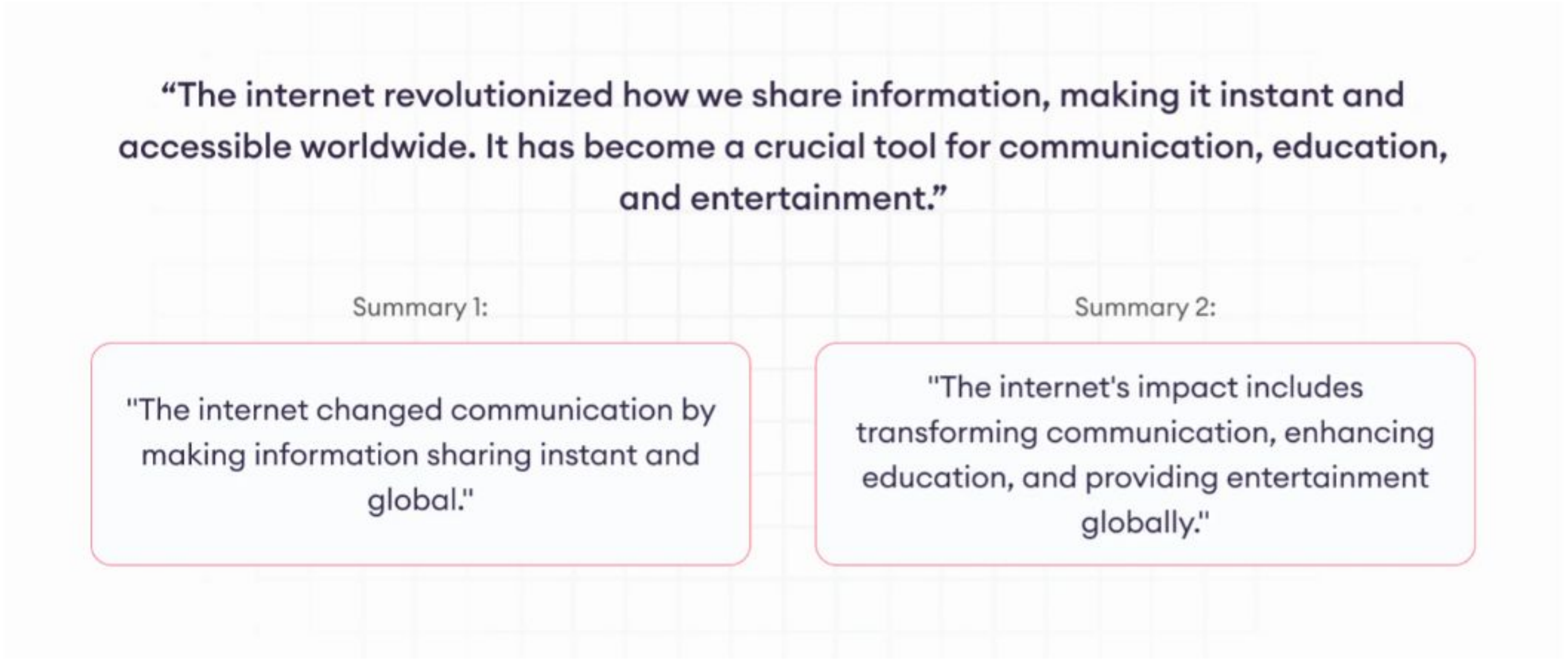
OPTIONS:

-yes -it is not possible to tell -no

**FLAN Response**

It is not possible to tell

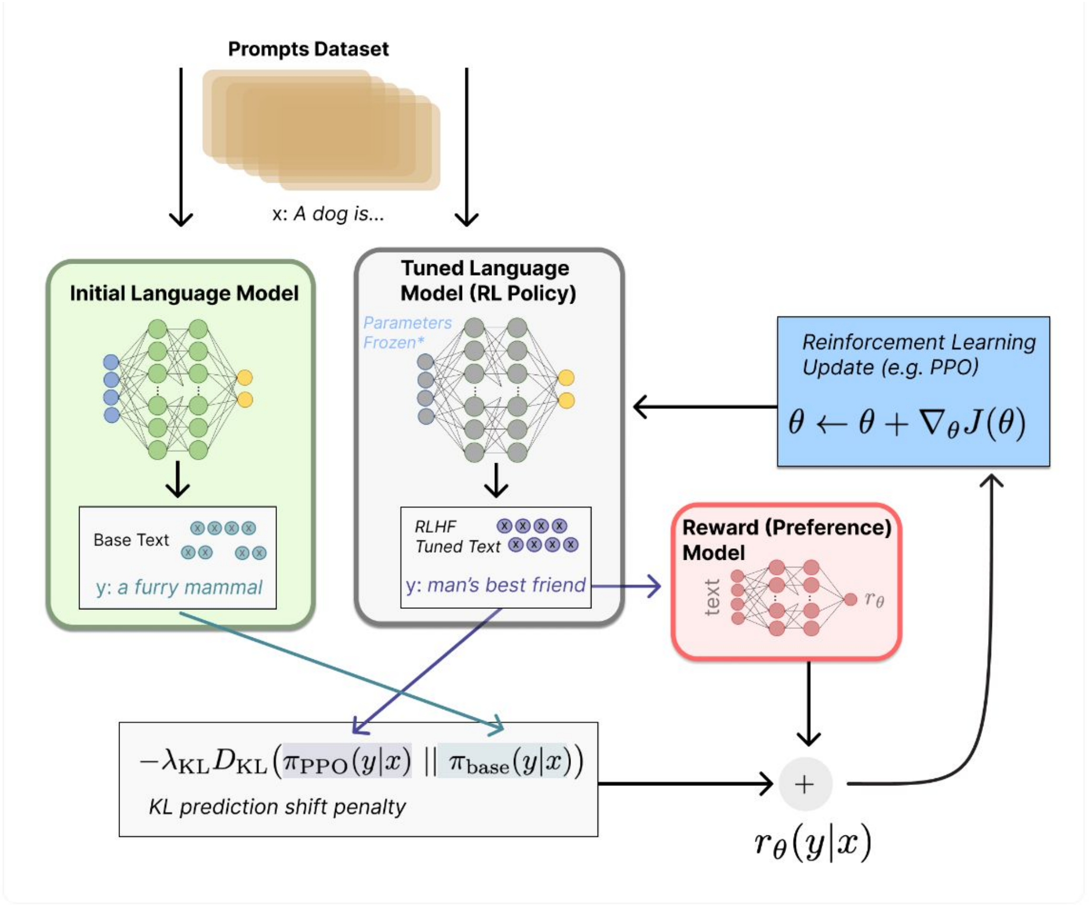
# What is aligning an LLM to Human Preferences?



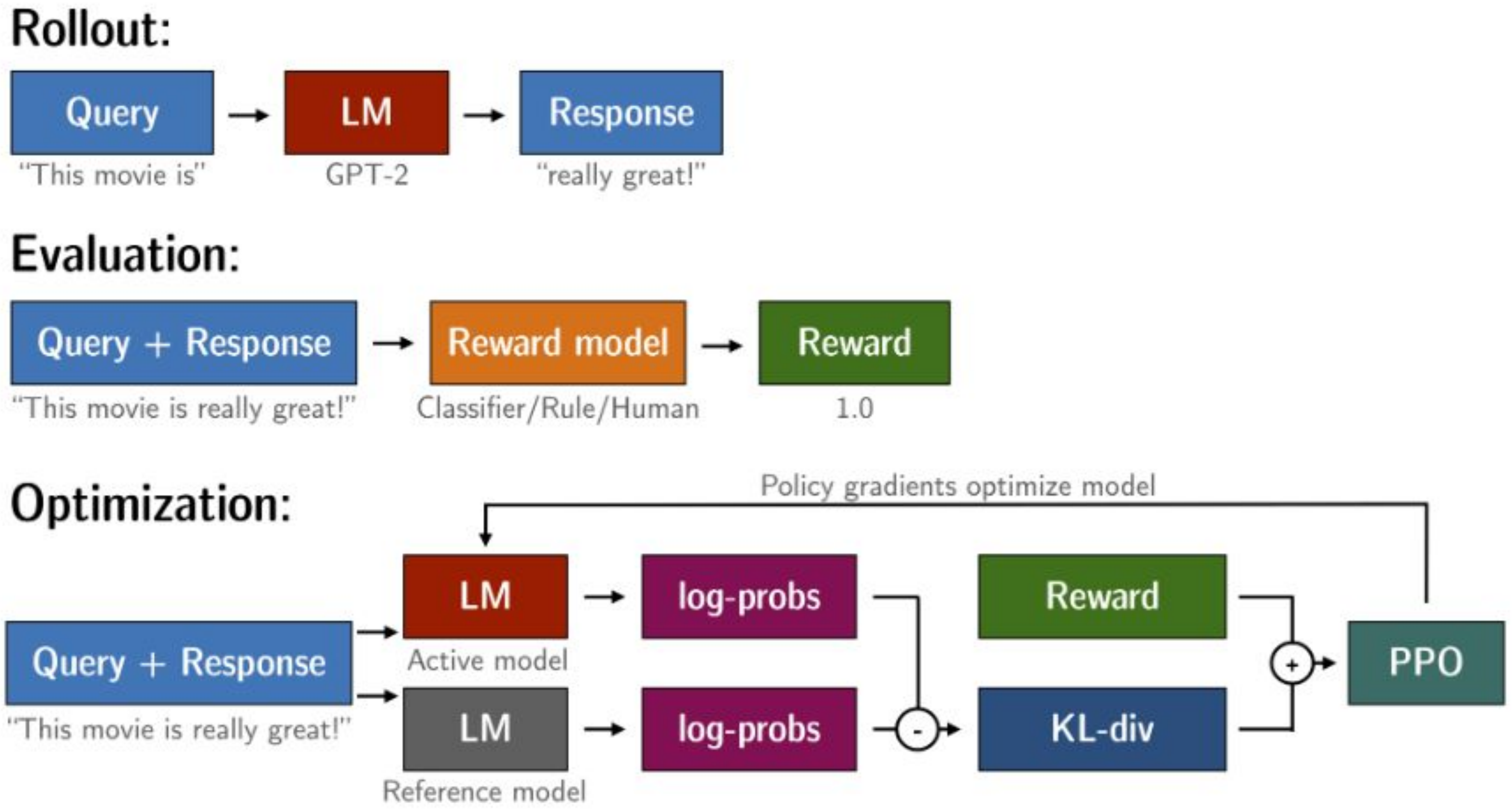
"The internet revolutionized how we share information, making it instant and accessible worldwide. It has become a crucial tool for communication, education, and entertainment."

| Summary 1: | Summary 2: |
|---|---|
| "The internet changed communication by making information sharing instant and global." | "The internet's impact includes transforming communication, enhancing education, and providing entertainment globally." |

# What is RLHF and PPO?


Prompts Dataset
x: A dog is...
Initial Language Model
Base Text
y: a furry mammal
Tuned Language Model (RL Policy)
Parameters Frozen*
RLHF Tuned Text
y: man's best friend
Reinforcement Learning Update (e.g. PPO)
$\theta \leftarrow \theta + \nabla_\theta J(\theta)$
Reward (Preference) Model
text
$r_\theta$
$-\lambda_{KL} D_{KL} \left(\pi_{PPO}(y|x) \;||\; \pi_{base}(y|x)\right)$
KL prediction shift penalty
+
$r_\theta(y|x)$

# Example - Align GPT-2 to generate positive content



# Example - Align GPT-2 to generate positive content

We'll walk through the process of making GPT-2 generate positive movie reviews.

## 1. Rollout - Standard Generation Process during training \ fine-tuning

- **Query:** You start with an incomplete movie review, for example, "This movie is."
- **Language Model (LM):** GPT-2 takes this input and generates a continuation, such as "really great!"
- **Response:** The generated continuation ("really great!") is what GPT-2 suggests as the completion of the review.

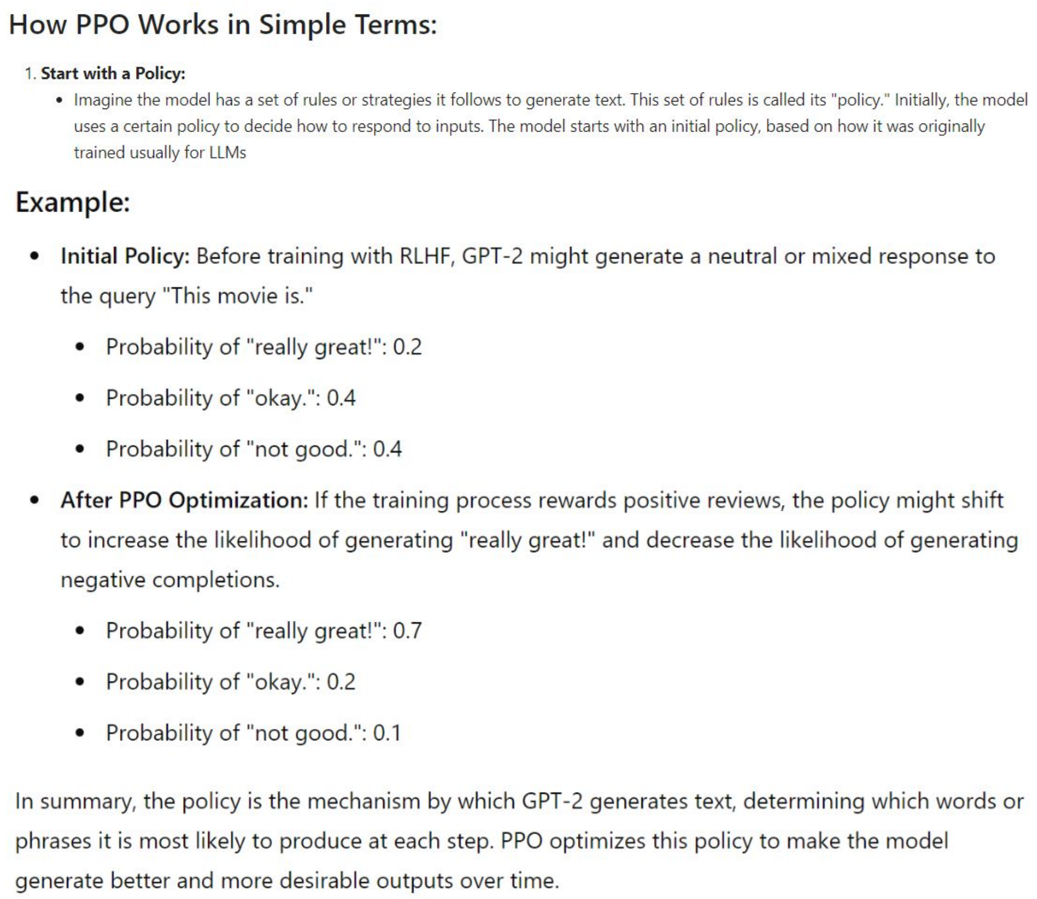
## 2. Evaluation

- **Query + Response:** The completed sentence, "This movie is really great!" is then evaluated.
- **Reward Model:** A separate model, called the reward model, assesses the quality of the response. It might be a classifier (like our BERT classifier), a set of rules, or even manual human feedback that judges whether the review is positive and well-formed.
- **Reward:** The reward model assigns a numerical score (reward) to the response based on how well it meets the desired criteria (in this case, being a positive review can get a high positive score >3 and negative review can get a low score < 1).

## 3. Optimization

- **Comparison with a Reference Model:** The GPT-2 model (model being fine-tuned and aligned) that generated the response is compared to a baseline or reference model, which can be an earlier, unoptimized version of GPT-2.
- **Log-Probabilities:** The likelihood (log-probabilities) of the generated response is calculated for both the active and reference models.
- **KL-Divergence (KL-div):** The difference between these probabilities (how much the active model's behavior deviates from the reference) is measured using KL-divergence.
- **PPO (Proximal Policy Optimization):** This is an algorithm that uses the reward and the difference in log-probabilities to adjust the model's parameters. The goal is to improve the model so it generates responses that receive higher rewards more consistently.
- **Final Update:** The model is optimized using policy gradients, balancing the reward and the deviation from the reference model to make GPT-2 better at generating positive reviews.

# Example - Align GPT-2 to generate positive content



## How PPO Works in Simple Terms:

1. **Start with a Policy:**
    - Imagine the model has a set of rules or strategies it follows to generate text. This set of rules is called its "policy." Initially, the model uses a certain policy to decide how to respond to inputs. The model starts with an initial policy, based on how it was originally trained usually for LLMs

### Example:

- **Initial Policy:** Before training with RLHF, GPT-2 might generate a neutral or mixed response to the query "This movie is."
    - Probability of "really great!": 0.2
    - Probability of "okay.": 0.4
    - Probability of "not good.": 0.4
- **After PPO Optimization:** If the training process rewards positive reviews, the policy might shift to increase the likelihood of generating "really great!" and decrease the likelihood of generating negative completions.
    - Probability of "really great!": 0.7
    - Probability of "okay.": 0.2
    - Probability of "not good.": 0.1

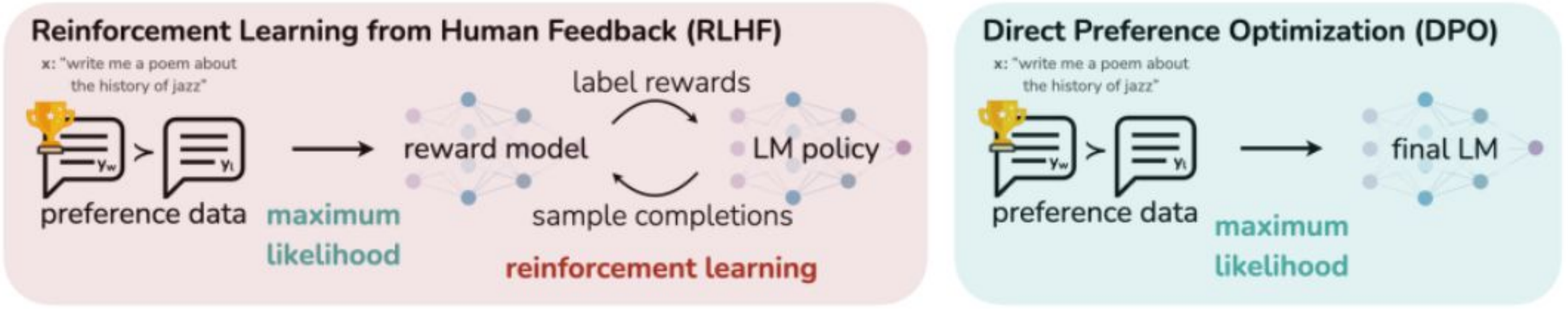
In summary, the policy is the mechanism by which GPT-2 generates text, determining which words or phrases it is most likely to produce at each step. PPO optimizes this policy to make the model generate better and more desirable outputs over time.

## Example - Align GPT-2 to generate positive content

2. **Evaluate the Policy:**
   - The model's current policy generates some responses to given inputs, and these responses are evaluated using the reward model. This evaluation tells us how good or bad the responses are (i.e., how well they match what we want, such as generating positive reviews).
3. **Calculate the Advantage:**
   - PPO compares how much better or worse the new policy (after generating a response) is compared to an older, reference policy. This comparison is called the "advantage." If the new policy is better (higher reward), that's positive; if it's worse, that's negative.
4. **Make Small Adjustments:**
   - Instead of making big changes to the policy (which might make the model behave unpredictably), PPO makes small, safe adjustments. It ensures that the updates are "proximal" or close to the previous policy, meaning the model doesn't stray too far from what it already knows.
   - This is done by using a technique called **clipping**. Clipping limits how much the policy can change in one update, preventing drastic changes that could lead to poor performance.
5. **Balance Exploration and Exploitation:**
   - PPO balances between two things:
     - **Exploration** (trying new ways of generating responses) and
     - **Exploitation** (sticking with what is already known to work well).
   - By carefully controlling updates, PPO allows the model to gradually improve its policy without making risky or unstable changes.
6. **Iterate:**
   - This process of making small updates, evaluating, and adjusting is repeated many times. Over time, the model's policy becomes more and more refined, leading to better performance (in this case, generating more positive and higher-quality reviews).

# RLHF free Alignment Methods - DPO

The challenge with RLHF is that it is a complex and often unstable procedure, first fitting a reward model that reflects the human preferences, and then fine-tuning the LLM using reinforcement learning to maximize this estimated reward without drifting too far from the original model using KL-divergence.


Reinforcement Learning from Human Feedback (RLHF)
x: "write me a poem about the history of jazz"
$y_w$ > $y_l$
preference data
maximum likelihood
reward model
label rewards
sample completions
LM policy
reinforcement learning
Direct Preference Optimization (DPO)
x: "write me a poem about the history of jazz"
$y_w$ > $y_l$
preference data
maximum likelihood
final LM

DPO optimizes for human preferences while avoiding reinforcement learning. Existing methods for fine-tuning language models with human feedback first train a reward model to a dataset of prompts and human preferences over pairs of responses (or supervised models like LLM classifiers or rankers),

and then use RL to find a policy that maximizes the learned reward.

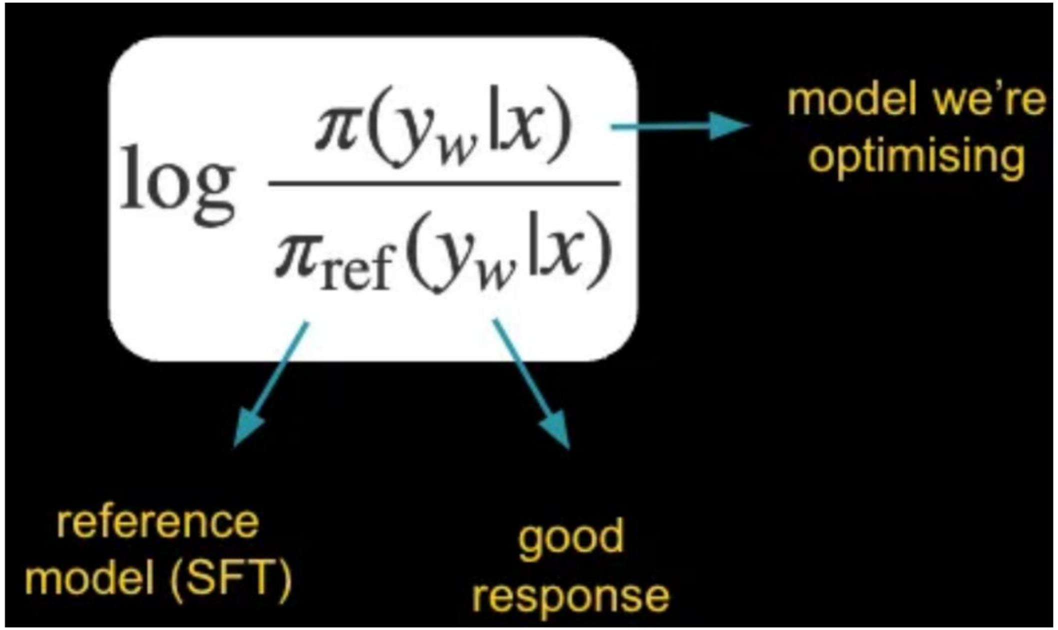
In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an implicit reward model whose corresponding optimal policy can be extracted in closed form.

# RLHF free Alignment Methods - DPO

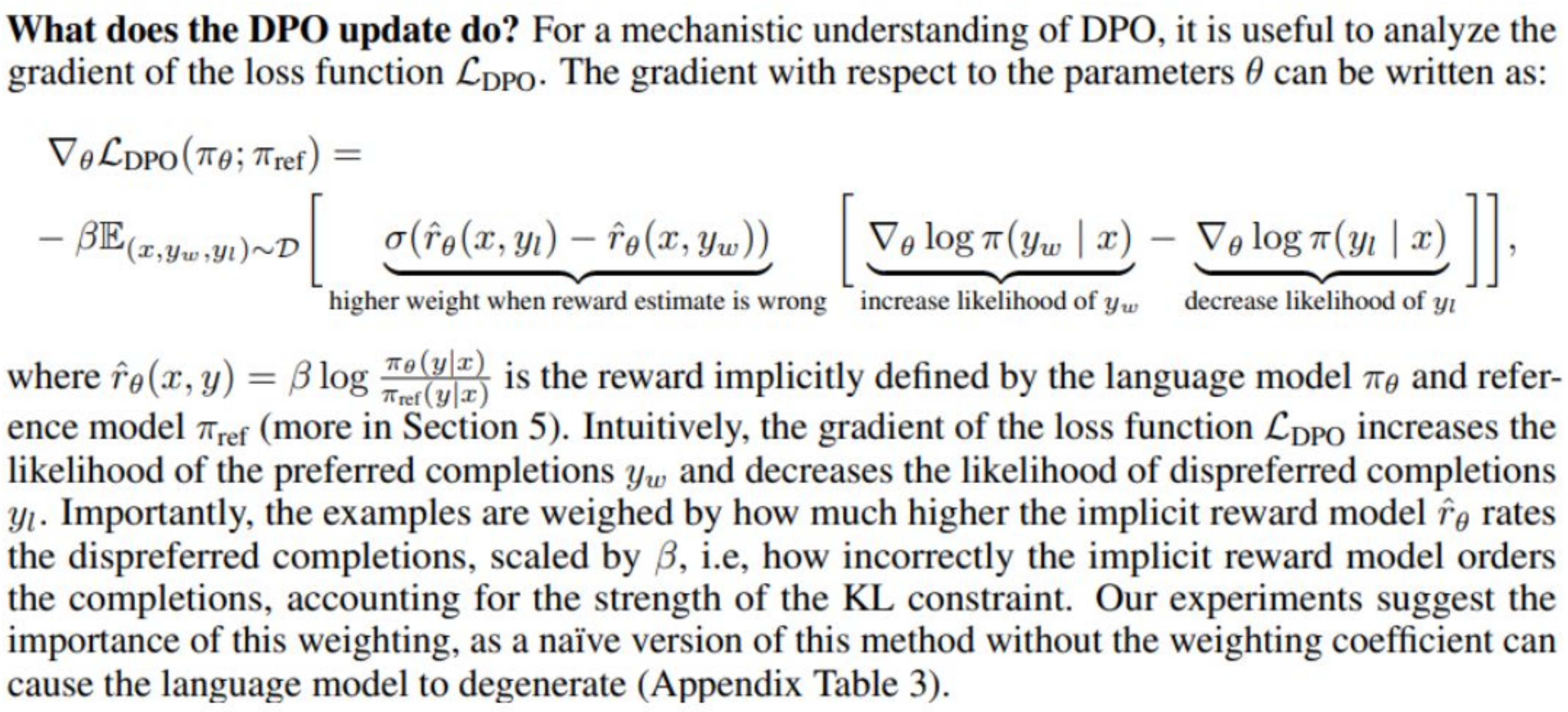
Conisdering a Dataset of human preferences {(x,yw,yl)}, where x is a prompt and yw, yl are the preferred and dis-preferred responses. The policy for DPO can be framed as:

$$\max_{\pi} \mathbb{E}_{(x,y_w,y_l)\sim D} \log \sigma \left( \beta \log \frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right)$$

Here the key aspects in the above equation include:



## RLHF free Alignment Methods - DPO



**What does the DPO update do?** For a mechanistic understanding of DPO, it is useful to analyze the gradient of the loss function $\mathcal{L}_{\text{DPO}}$. The gradient with respect to the parameters $\theta$ can be written as:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$
$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \bigg[ \underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}} \bigg[ \underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l} \bigg] \bigg],$$

where $\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ is the reward implicitly defined by the language model $\pi_\theta$ and reference model $\pi_{\text{ref}}$ (more in Section 5). Intuitively, the gradient of the loss function $\mathcal{L}_{\text{DPO}}$ increases the likelihood of the preferred completions $y_w$ and decreases the likelihood of dispreferred completions $y_l$. Importantly, the examples are weighed by how much higher the implicit reward model $\hat{r}_\theta$ rates the dispreferred completions, scaled by $\beta$, i.e, how incorrectly the implicit reward model orders the completions, accounting for the strength of the KL constraint. Our experiments suggest the importance of this weighting, as a naïve version of this method without the weighting coefficient can cause the language model to degenerate (Appendix Table 3).

# RLHF free Alignment Methods - ORPO

The challenge with RLHF or DPO is you need to keep aligning the SFT model with a reference model to make sure it doesn't stray away too far. This is where Odds Ratio Preference Optimization (ORPO) can be quite useful!

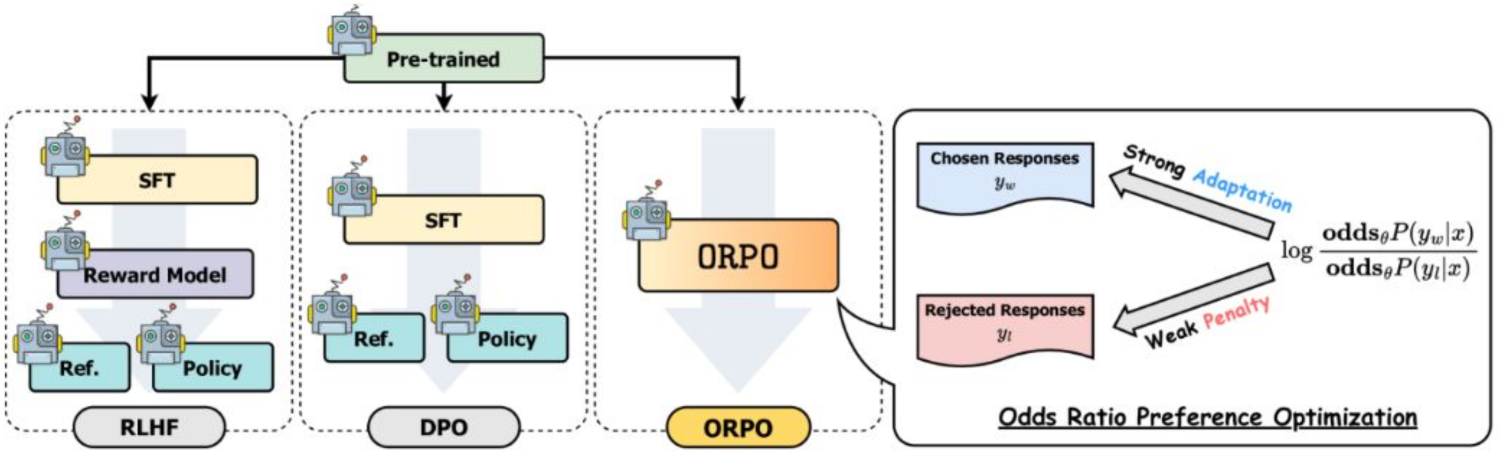
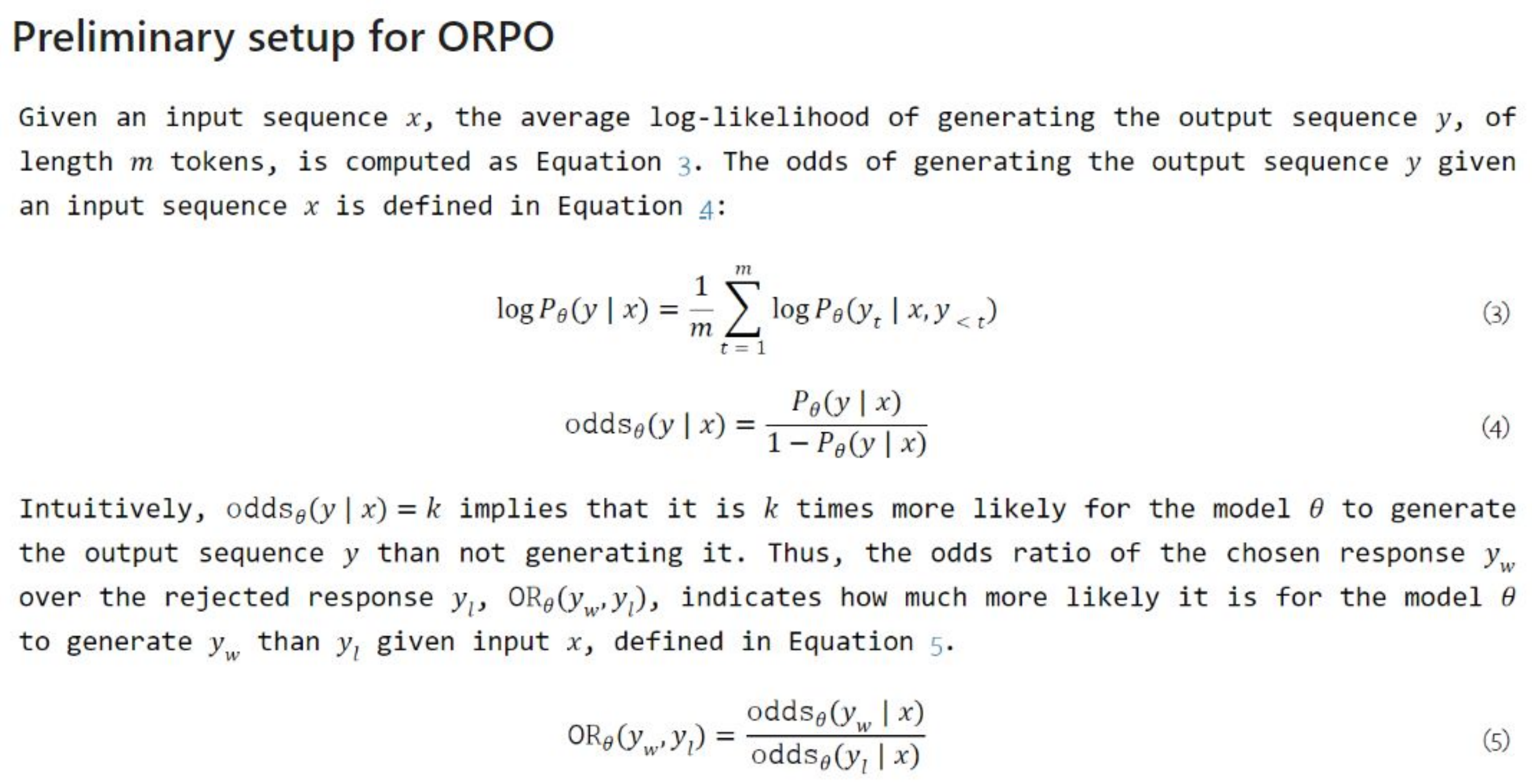


Figure 2: Comparison of model alignment techniques. ORPO aligns the language model *without a reference model* in a single-step manner by assigning a weak penalty to the rejected responses and a strong adaptation signal to the chosen responses with a simple log odds ratio term appended to the negative log-likelihood loss.

ORPO aligns the language model without a reference model in a single-step manner by assigning a weak penalty to the rejected responses and a strong adaptation signal to the chosen responses with a simple log odds ratio term appended to the negative log-likelihood loss.

# RLHF free Alignment Methods - ORPO



## Preliminary setup for ORPO

Given an input sequence $x$, the average log-likelihood of generating the output sequence $y$, of length $m$ tokens, is computed as Equation 3. The odds of generating the output sequence $y$ given an input sequence $x$ is defined in Equation 4:

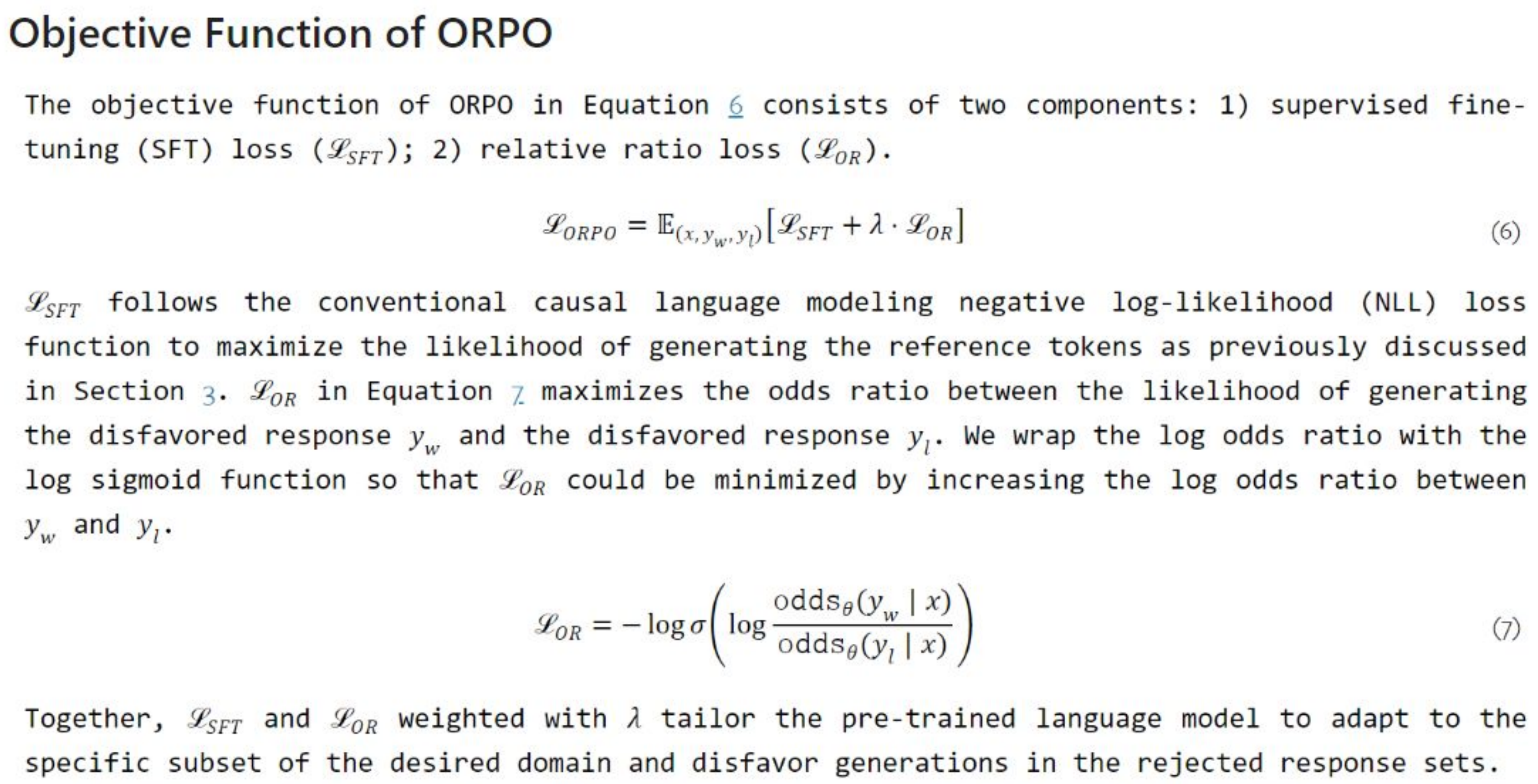
$$\log P_\theta(y \mid x) = \frac{1}{m}\sum_{t=1}^{m} \log P_\theta(y_t \mid x, y_{<t}) \tag{3}$$

$$\text{odds}_\theta(y \mid x) = \frac{P_\theta(y \mid x)}{1 - P_\theta(y \mid x)} \tag{4}$$

Intuitively, $\text{odds}_\theta(y \mid x) = k$ implies that it is $k$ times more likely for the model $\theta$ to generate the output sequence $y$ than not generating it. Thus, the odds ratio of the chosen response $y_w$ over the rejected response $y_l$, $\text{OR}_\theta(y_w, y_l)$, indicates how much more likely it is for the model $\theta$ to generate $y_w$ than $y_l$ given input $x$, defined in Equation 5.

$$\text{OR}_\theta(y_w, y_l) = \frac{\text{odds}_\theta(y_w \mid x)}{\text{odds}_\theta(y_l \mid x)} \tag{5}$$

# RLHF free Alignment Methods - ORPO



## Objective Function of ORPO

The objective function of ORPO in Equation 6 consists of two components: 1) supervised fine-tuning (SFT) loss ($\mathscr{L}_{SFT}$); 2) relative ratio loss ($\mathscr{L}_{OR}$).

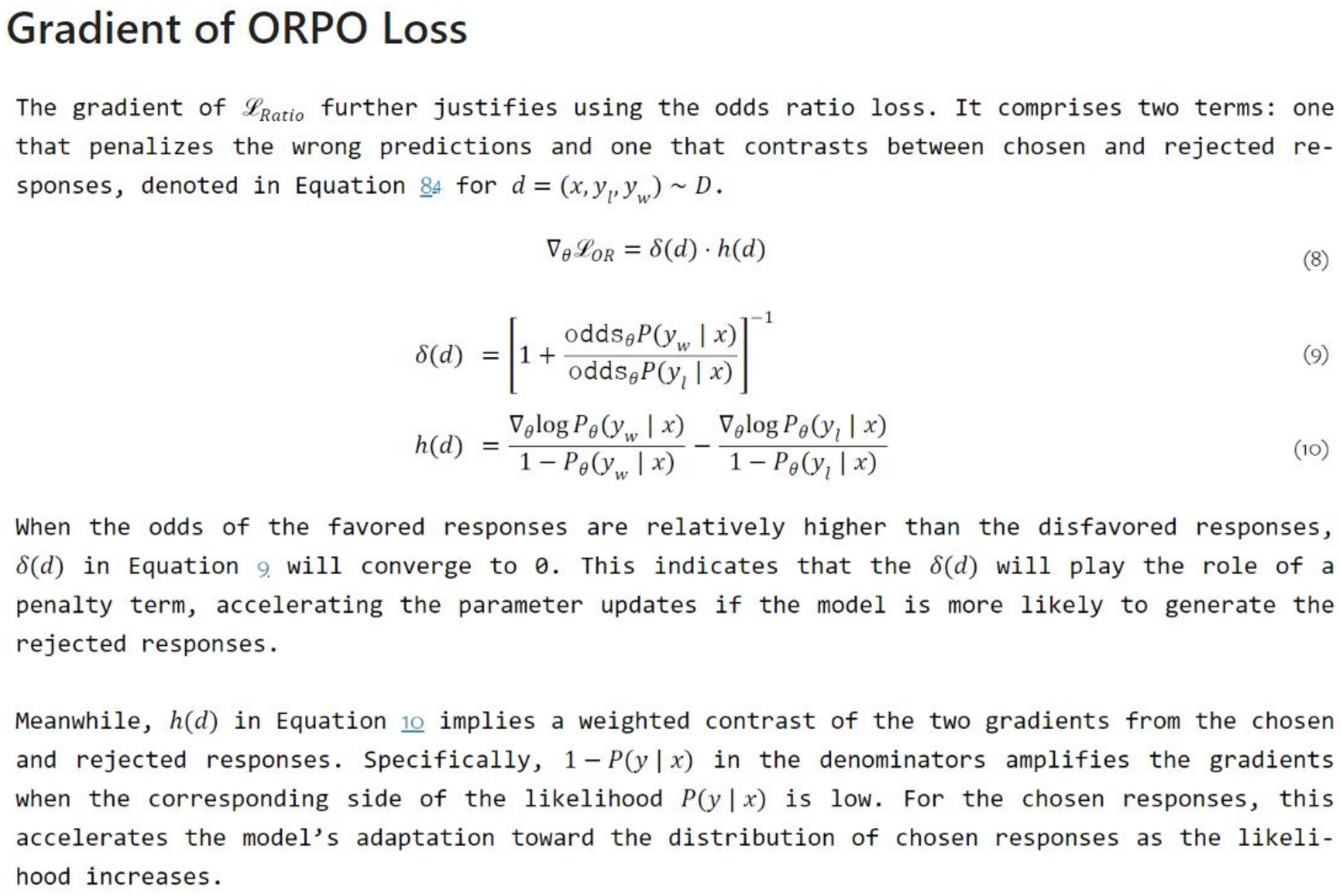
$$\mathscr{L}_{ORPO} = \mathbb{E}_{(x, y_w, y_l)}\left[\mathscr{L}_{SFT} + \lambda \cdot \mathscr{L}_{OR}\right] \tag{6}$$

$\mathscr{L}_{SFT}$ follows the conventional causal language modeling negative log-likelihood (NLL) loss function to maximize the likelihood of generating the reference tokens as previously discussed in Section 3. $\mathscr{L}_{OR}$ in Equation 7 maximizes the odds ratio between the likelihood of generating the disfavored response $y_w$ and the disfavored response $y_l$. We wrap the log odds ratio with the log sigmoid function so that $\mathscr{L}_{OR}$ could be minimized by increasing the log odds ratio between $y_w$ and $y_l$.

$$\mathscr{L}_{OR} = -\log \sigma \left( \log \frac{\mathrm{odds}_\theta(y_w \mid x)}{\mathrm{odds}_\theta(y_l \mid x)} \right) \tag{7}$$

Together, $\mathscr{L}_{SFT}$ and $\mathscr{L}_{OR}$ weighted with $\lambda$ tailor the pre-trained language model to adapt to the specific subset of the desired domain and disfavor generations in the rejected response sets.

# RLHF free Alignment Methods - ORPO



## Gradient of ORPO Loss

The gradient of $\mathscr{L}_{Ratio}$ further justifies using the odds ratio loss. It comprises two terms: one that penalizes the wrong predictions and one that contrasts between chosen and rejected responses, denoted in Equation 84 for $d = (x, y_l, y_w) \sim D$.

$$\nabla_\theta \mathscr{L}_{OR} = \delta(d) \cdot h(d) \tag{8}$$

$$\delta(d) = \left[1 + \frac{\text{odds}_\theta P(y_w \mid x)}{\text{odds}_\theta P(y_l \mid x)}\right]^{-1} \tag{9}$$

$$h(d) = \frac{\nabla_\theta \log P_\theta(y_w \mid x)}{1 - P_\theta(y_w \mid x)} - \frac{\nabla_\theta \log P_\theta(y_l \mid x)}{1 - P_\theta(y_l \mid x)} \tag{10}$$

When the odds of the favored responses are relatively higher than the disfavored responses, $\delta(d)$ in Equation 9 will converge to 0. This indicates that the $\delta(d)$ will play the role of a penalty term, accelerating the parameter updates if the model is more likely to generate the rejected responses.

Meanwhile, $h(d)$ in Equation 10 implies a weighted contrast of the two gradients from the chosen and rejected responses. Specifically, $1 - P(y \mid x)$ in the denominators amplifies the gradients when the corresponding side of the likelihood $P(y \mid x)$ is low. For the chosen responses, this accelerates the model's adaptation toward the distribution of chosen responses as the likelihood increases.

Analytics Vidhya