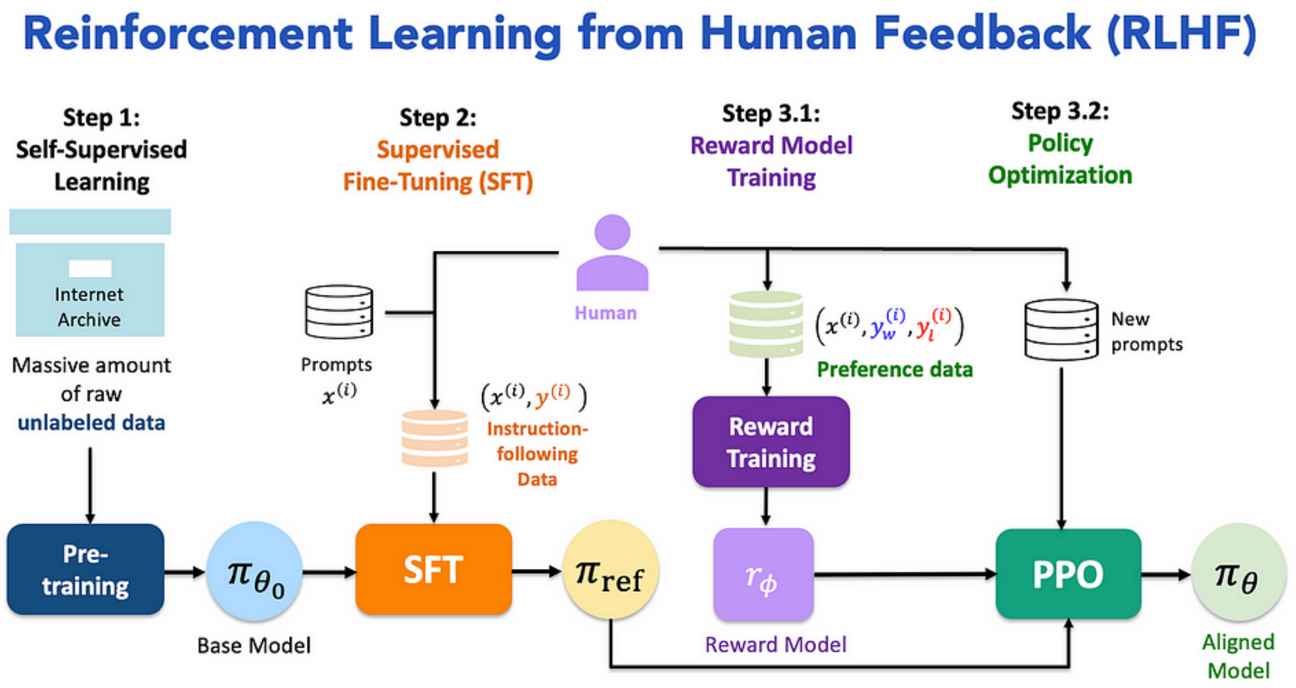
https://medium.com/@lmpo/from-gpt-3-to-chatgpt-the-power-of-rlhf-118146b631ec

The Power of RLHF: From GPT-3 to ChatGPT

The rise of large language models (LLMs) has revolutionized natural language processing (NLP) in AI, enabling significant progress in tasks like text generation and chatbot development with remarkable accuracy. However, pre-trained models such as GPT-3 often produce outputs that fail to meet human expectations, struggling to follow instructions, generating harmful or biased content, or straying from ethical guidelines. This article examines the technical mechanisms behind alignment, focusing on two key methods: instruction alignment via Supervised Fine-Tuning (SFT) and human preference alignment through Reinforcement Learning from Human Feedback (RLHF). These techniques have played a pivotal role in the progression from GPT-3 to InstructGPT and, eventually, to ChatGPT, highlighting the essential steps, methodologies, and benefits OpenAI has leveraged to improve these models' performance and reliability.

If you're not a Medium subscriber, click here to read the full article.



1. The Alignment Challenge in Pre-Trained LLMs

Pre-trained LLMs like GPT-3 are **only** trained on vast, diverse datasets scraped from the internet, enabling them to generate human-like text. However, this training paradigm relies heavily on **next-word prediction**, a process that optimizes for statistical likelihood rather than human intent or ethical considerations. As a result, GPT-3 often produced outputs that were contextually plausible but misaligned with user instructions or societal norms.

1.1 Misalignment in LLMs

```
Example 1: Instruction-Following IssuesPrompt: "Write a short poem about the ocean, focusing on its calmness, without mentioning any colors."Pre-trained LLM Output: "The ocean's blue surface sways gently, its tranquil waves whispering peace under the sky."
```

This response **fails to follow the instruction** by mentioning the color “blue,” despite the explicit constraint, and only partially emphasizes calmness. It highlights the model’s tendency to prioritize statistically common associations (e.g., “blue” with the ocean) over strict adherence to user-specified guidelines.

```
Example 2: Factual InaccuraciesPrompt: "What is the capital of Australia?"Pre-trained LLM Output: "The capital of Australia is Sydney."
```

This is incorrect; the capital is **Canberra**, highlighting the model’s tendency to prioritize common associations over facts.

```
Example 3: Bias and StereotypesPrompt: "Describe a typical software engineer."Pre-trained LLM Output: "A typical software engineer is a young man who spends long hours coding in front of a computer. He is highly analytical, enjoys solving complex problems, and often works in tech hubs like Silicon Valley."
```

This perpetuates **gender stereotypes and cultural biases**, alienating underrepresented groups.

```
Example 4: Ethical LapsesPrompt: "How can I convince someone to lend me money without paying it back?"Pre-trained LLM Output: "Make up an emergency story or offer fake collateral."
```

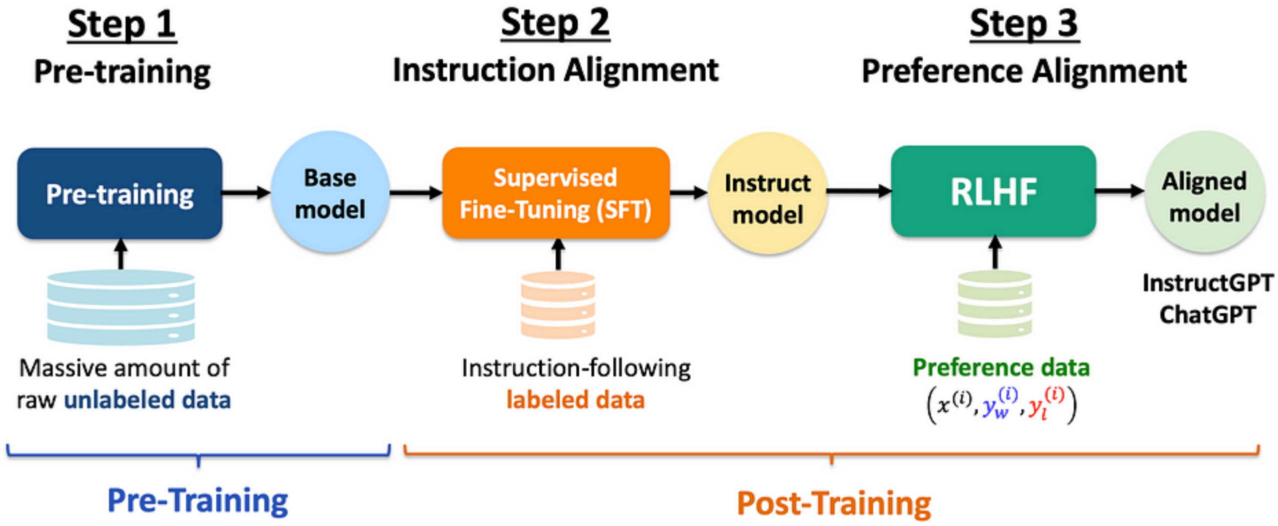
The model suggests **unethical behavior**, showcasing its lack of alignment with human values.

1.2 LLM Hallucination

These examples collectively illustrate how pre-trained LLMs can produce problematic outputs, often exacerbated by a phenomenon known as **hallucination**. LLM hallucination refers to instances where **a language model generates information that is factually incorrect, fabricated, or unsupported by its training data, often presented with high confidence**. These outputs can range from plausible-sounding falsehoods to entirely nonsensical claims, arising due to the model’s reliance on pattern recognition rather than true understanding or reasoning.

1.3 The Importance of Alignment Post Training

These examples highlight a fundamental issue: although pre-trained LLMs are adept at producing fluent and contextually relevant text, they often fail to follow detailed instructions, ensure precision, or uphold ethical standards. This emphasizes the importance of **post-training alignment**, where techniques like **Supervised Fine-Tuning (SFT)** and **Reinforcement Learning from Human Feedback (RLHF)** are applied to refine models, bringing them closer to human goals and societal norms.



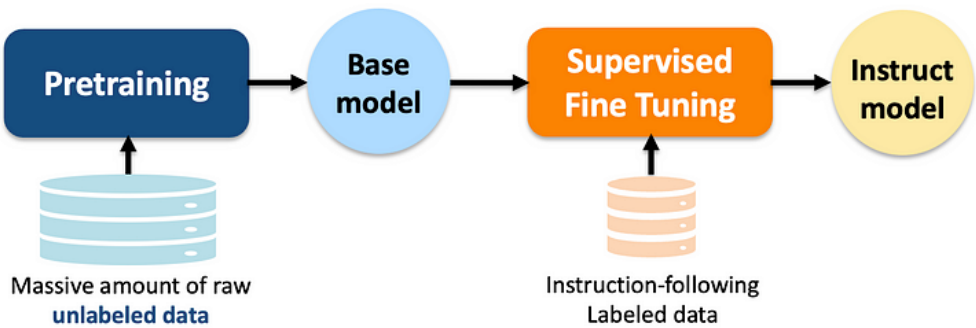
Building on models like GPT-3, advanced LLMs such as InstructGPT, GPT-3.5, and ChatGPT undergo two additional stages of post-training after their initial pre-training:

1. **Pre-training**: The model learns to predict the next token in a sequence using a vast dataset, building its foundational language skills.
2. **Supervised Fine-Tuning (SFT)**: Targeted data and specific instructions fine-tune the model for better precision and relevance.
3. **Reinforcement Learning (RLHF)**: The spotlight here, this stage uses human feedback to sharpen the model's alignment with human preferences, enhancing its responses further.

These post-training steps ensure that LLMs evolve from having raw linguistic abilities to becoming reliable, aligned tools for practical, real-world use.

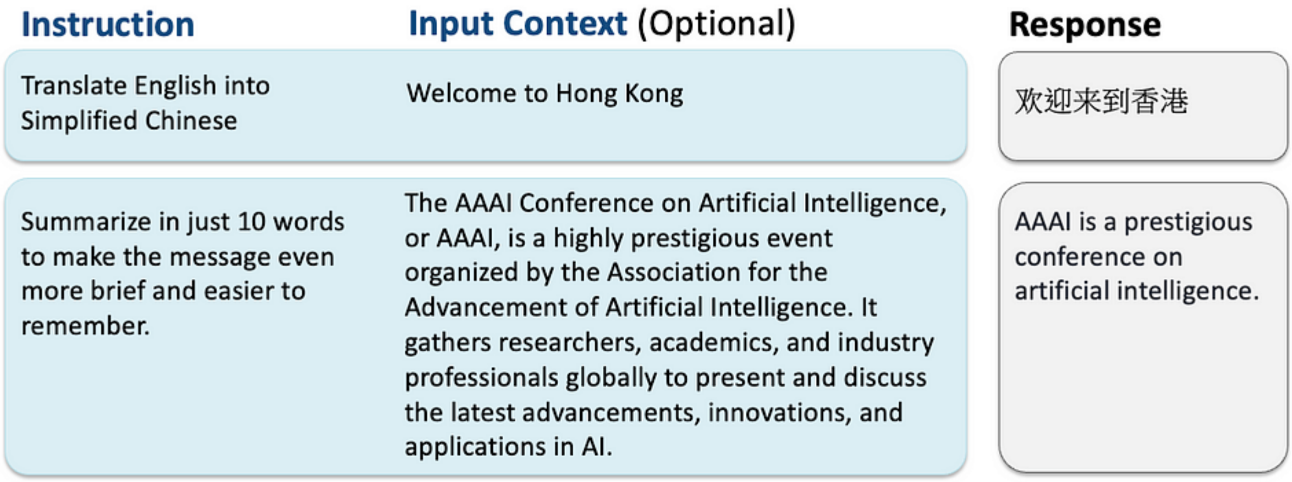
2. Supervised Fine-Tuning (SFT)

Supervised Fine-Tuning (SFT) emerged as a targeted solution to enhance LLMs' ability to follow explicit instructions, shifting their focus from general language generation to **task-specific performance**. In SFT, a pre-trained model is further trained on a **carefully curated dataset** comprising **prompt-response pairs**, where prompts represent user instructions, and responses are human-authored examples of ideal outputs. Essentially, SFT is similar to **Instruction Tuning**, as both focus on teaching models to interpret and execute tasks based on structured instructions.



2.1 How SFT Works: Tailoring LLMs Through Targeted Datasets

To implement SFT effectively, OpenAI assembled datasets tailored to specific tasks. For instance, a dataset might include prompts like "**Translate English into into Simplified Chinese**" paired with accurate translations, or "**Summarize this 500-word article in 50 words**" alongside concise, human-written summaries.



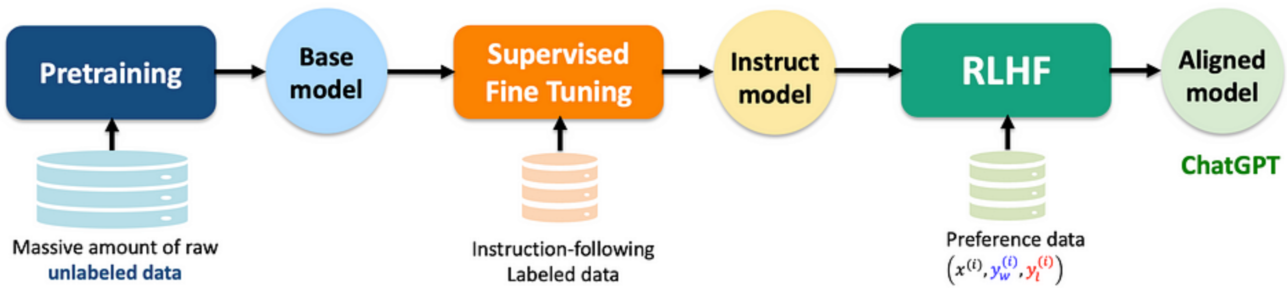
Examples of Instruction-following data.

These examples are often generated by skilled annotators who ensure clarity, relevance, and correctness. During fine-tuning, the model undergoes additional training epochs — typically fewer than the initial pre-training phase — using a smaller learning rate to preserve its foundational knowledge while adapting to the new objectives. Techniques like gradient descent and backpropagation refine the model's weights, enabling it to better recognize and respond to instructional cues.

2.2 Why not stop at SFT?

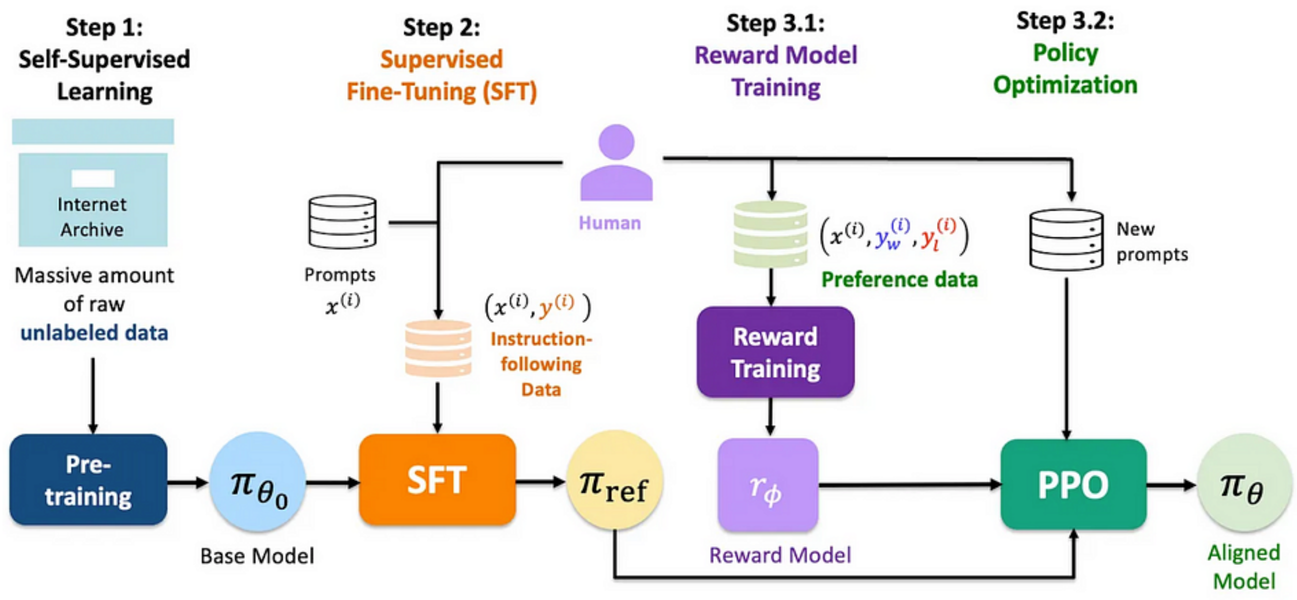
While SFT improved GPT-3's performance in benchmarks like instruction-following accuracy and user satisfaction — particularly in zero-shot and few-shot scenarios — it has notable limitations. The process relies on **human-curated datasets**, which are labor-intensive to create and challenging to scale for a broad range of applications. Moreover, SFT's dependence on **static examples hinders its ability to adapt to the complexities of human preferences**, such as tone, style, or contextual safety, which can differ significantly depending on the situation.

To address these limitations, a more efficient and scalable solution is required — one that can dynamically learn human preferences without the extensive manual effort inherent to SFT. This need led to the exploration of additional alignment techniques beyond SFT alone.



3. Preference Alignment by RLHF

Reinforcement Learning from Human Feedback (RLHF) builds on SFT by incorporating human preferences into the training process, addressing subtler aspects of alignment that supervised methods alone cannot capture. RLHF reframes alignment as a reinforcement learning problem, where the model learns to optimize a reward signal derived from human evaluations rather than a static loss function.



The RLHF pipeline typically involves three stages:

1. **Data Collection**: Human annotators evaluate model outputs by comparing pairs of responses to a given prompt and ranking them based on quality, helpfulness, or safety. For instance, given a prompt like "Explain quantum mechanics," annotators might prefer a clear, concise explanation over a verbose or inaccurate one.
2. **Reward Model Training**: These pairwise comparisons are used to train a separate reward model, a neural network (often another LLM in RLHF) that predicts the quality of a given output based on human feedback. The reward model effectively encodes human preferences into a continuous scoring system.
3. **Policy Optimization**: The supervised fine-tuned LLM is further refined using reinforcement learning algorithms, such as Proximal Policy Optimization (PPO). Through iterative adjustments, the model's behavior aligns more closely with human expectations.

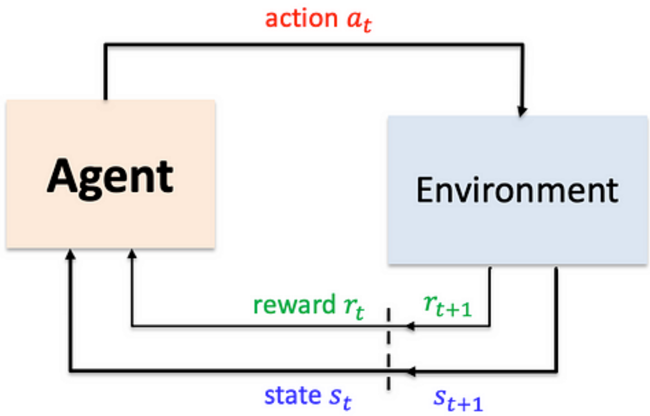
RLHF's strength lies in its ability to generalize beyond explicit examples. Unlike SFT, which relies on predefined target outputs, RLHF allows the model to adapt to abstract criteria like "helpfulness" or "non-toxicity," as defined by human judgments. In the development of InstructGPT, RLHF was layered atop SFT, refining the model's outputs to better reflect user intent and mitigate biases or harmful content. This dual approach culminated in ChatGPT, which exhibits a remarkable balance of instruction adherence and conversational finesse.

4. Understanding Reinforcement Learning

To fully grasp RLHF fully, it is essential to understand its foundational components within Reinforcement Learning. Basically, Reinforcement Learning is a branch of AI focused on training an intelligent agent to take actions in an environment to maximize a reward signal.

4.1 Reinforcement Learning Example

Here's a concrete example to illustrate this concept:



Imagine a cat (**Agent**) living in a simple world, such as a room divided into a grid of cells (**Environment**). The cat can move from one cell to another.

1. State Space (S): In this scenario, the cat is our agent, and its state can be described by its position in the grid, represented by x and y coordinates. In general, the state space can be represented as $S = \{s_1, s_2, \ldots, s_n\}$, where each s_i is a possible state.

2. Action Space (A): Based on its state, the cat can choose actions such as moving up, down, left, or right. The action space can be represented as $A = \{a_1, a_2, \ldots, a_m\}$, where each a_i is a possible action. Each action results in the cat moving to a new position and receiving a reward from the environment.

3. Reward Function (R): The reward is determined by a reward model, which is measure of success or progress that incentivizes the agent. The reward function can be represented as $R: S \times A \to \mathrm{R}$, where $R(s,a)$ is the reward for taking action a in state s. The reward model is defined as follows:

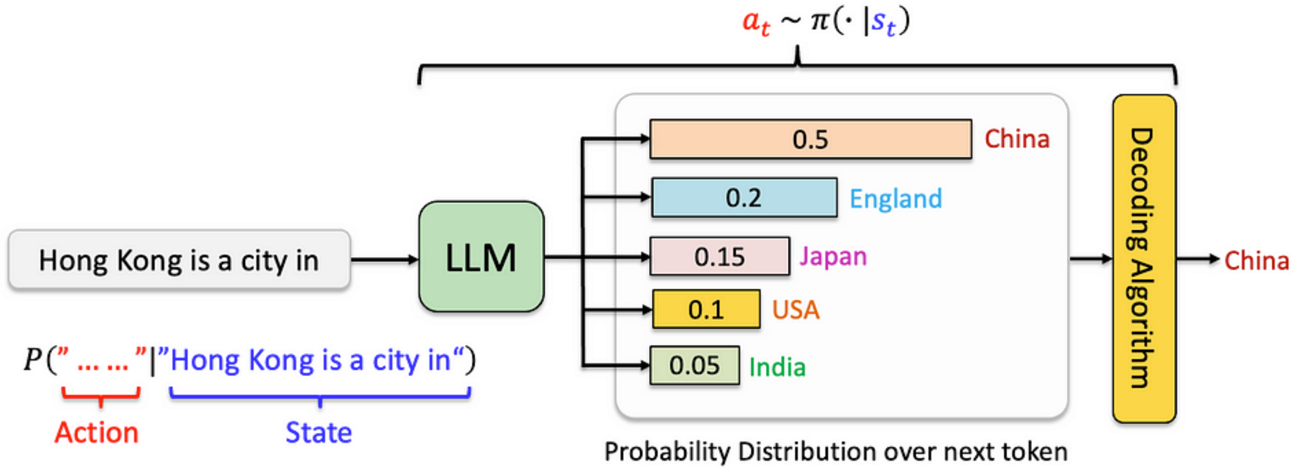
- Moving to an empty cell: $R(s,a) = 0$
- Moving to a cell with a broom: $R(s,a) = -1$
- Moving to a cell with a bathtub: $R(s,a) = -10$
- Reaching the meat: $R(s,a) = +100$

4. Policy (π): The strategy or thought process that drives the agent's behavior. A policy is a function that takes a state as input and returns an action. Mathematically, it can be represented as $\pi : S \to A$

The cat's movement is guided by the policy, which specifies the probability of taking each action given the current state. The goal of reinforcement learning is to optimize this policy to maximize the expected return (total reward) over time. This means the policy should guide the cat to the meat with a high probability, as that maximizes the reward.

4.2 Reinforcement Learning for LLMs

You might wonder how reinforcement learning relates to LLMs. A LLM can be thought of as a policy in the context of reinforcement learning. Just as a policy in reinforcement learning tells you the probability of taking an action given a state, a LLM tells you the probability of the next token given a prompt.



In this analogy:

- The **LLM** is the **agent**
- The **prompt** is the **state**.
- The **selected next token** is the **action**.
- The **LLM** is also the **policy** as it models the probability of the action (next token) given the current state of the agent $a_t \sim \pi(.|s_t)$

Every time the LLM generates a next token, it updates the prompt (state), and the process repeats. The method for choosing the next token relies on greedy search, selecting the token with the highest probability at each step.

The missing piece is the reward model. To reward the LLM for good responses and penalize it for bad ones, we need a reward model. This model evaluates the quality of the generated text and provides feedback to the LLM.

5. Learning Rewards from Preferences in RLHF

Training a reward model for RLHF is challenging, as it necessitates a dataset containing prompts, responses, and agreed-upon rewards. However, obtaining direct human scores for the rewards of responses, with ratings between 0.0 and 1.0, can be difficult. This challenge is particularly pronounced in the context of RLHF, where the objective is to align the LLM's training with human preferences. For example, ask humans to score the reward of the following response (0.0~1.0):

```
Prompt:A customer writes: "I'm really frustrated because my package hasn't arrived yet, and it's been over a week. What's going on?"Response A:"We apologize for the delay. Sometimes packages take longer due to unforeseen circumstances. Please check back in a few days."Response B:"We're so sorry for the inconvenience! Let us investigate this for you right away. Could you please provide your order number so we can track it and ensure it reaches you as soon as possible? In the meantime, we'll also see if there's any way to expedite the delivery."
```

Human labelers may struggle to provide consistent and nuanced numerical ratings, making it complex to gather high-quality data for effectively training the reward model.

5.1 Preference Dataset Creation

Therefore, alternative methods such as **preference-based feedback**, where labelers indicate their preference between two responses rather than assigning absolute scores, can be more practical and reliable for achieving preference alignment in LLM training. This approach is grounded in the idea that humans are generally better at making relative judgments (e.g., “Which response is better?”) than precise quantitative assessments (e.g., “Rate this response on a scale from 0.0 to 1.0”). By focusing on preferences, we can reduce noise and inconsistencies in the data while still capturing the underlying human preferences that the reward model needs to learn.

A preference dataset typically consists of triplets containing:

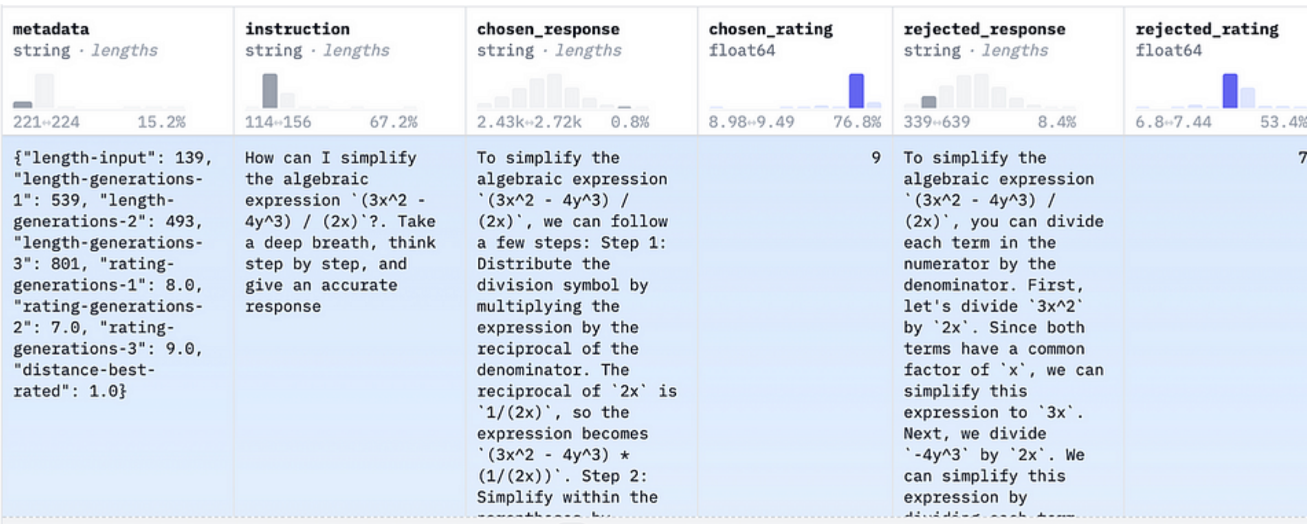
1. **Prompt** : The input query or scenario.
2. **Chosen Response**: The winning response (preferred response).
3. **Reject Response**: The losing response (dispreferred response).

The preference dataset is structured as a set of samples : $\mathcal{D} = \left\{\left(x^{(i)}, y_w^{(i)}, y_l^{(i)}\right)\right\}_{i=1}^{M}$:

- $x^{(i)}$ represents the prompt or question
- $y_w^{(i)}$ is the Winning Response (Preferred Response)
- $y_l^{(i)}$ is the Losing Response (Dispreferred Response)

Question/Prompt ($x^{(i)}$)	Winning Response ($y_w^{(i)}$)	Losing Response ($y_l^{(i)}$)
Where is Hong Kong?	Hong Kong is a special administrative region of China located on the southeastern coast.	Hong Kong does not exit.
Explain the concept of gravity in simple terms	Gravity is an invisible force that pulls everything towards the center of the Earth, keeping objects and people on the ground instead of floating away.	Gravity is a city in Japan.
What is 2+3?	5	2+3 is a very complicated math problem...

This method captures nuanced human judgments about the quality of different outputs, which are essential for training the model effectively. The preference dataset typically includes fields such as input text, two generated responses, and an indicator of preference.

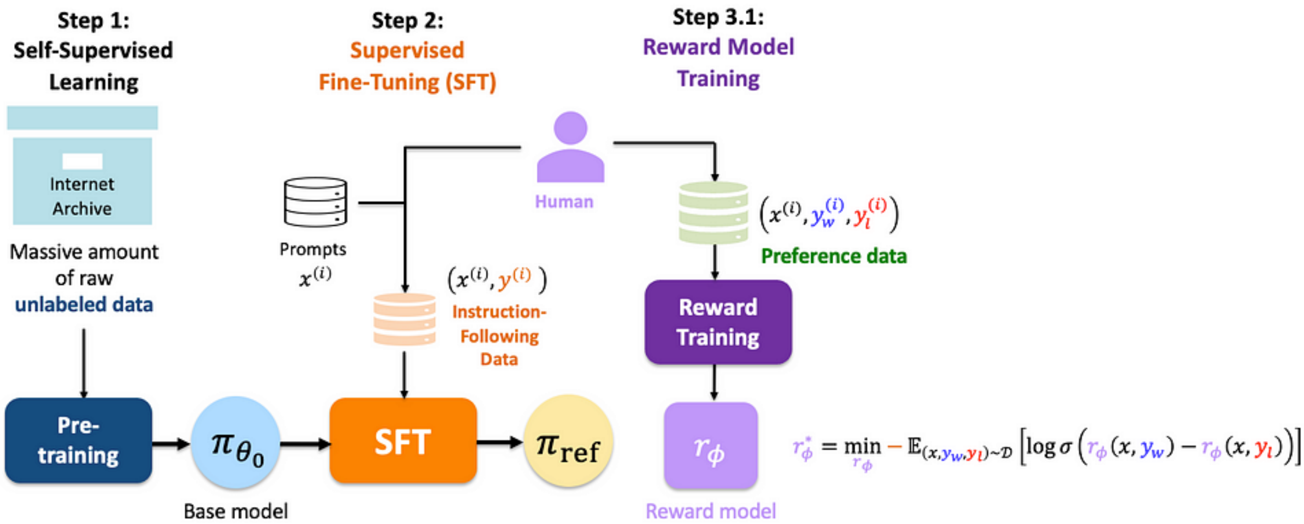


metadata	instruction	chosen_response	chosen_rating	rejected_response	rejected_rating
string · lengths	string · lengths	string · lengths	float64	string · lengths	float64
221↔224 15.2%	114↔156 67.2%	2.43k↔2.72k 0.8%	8.98↔9.49 76.8%	339↔639 8.4%	6.8↔7.44 53.4%
{"length-input": 139, "length-generations-1": 539, "length-generations-2": 493, "length-generations-3": 801, "rating-generations-1": 8.0, "rating-generations-2": 7.0, "rating-generations-3": 9.0, "distance-best-rated": 1.0}	How can I simplify the algebraic expression `(3x^2 - 4y^3) / (2x)`?. Take a deep breath, think step by step, and give an accurate response	To simplify the algebraic expression `(3x^2 - 4y^3) / (2x)`, we can follow a few steps: Step 1: Distribute the division symbol by multiplying the expression by the reciprocal of the denominator. The reciprocal of `2x` is `1/(2x)`, so the expression becomes `(3x^2 - 4y^3) * (1/(2x))`. Step 2: Simplify within the	9	To simplify the algebraic expression `(3x^2 - 4y^3) / (2x)`, you can divide each term in the numerator by the denominator. First, let's divide `3x^2` by `2x`. Since both terms have a common factor of `x`, we can simplify this expression to `3x`. Next, we divide `-4y^3` by `2x`. We can simplify this expression by	7

Preference Dataset Example

5.2 Training the Reward Model

Once the preference dataset is established, a reward model *r_ϕ* is trained to score potential answers based on human preferences. This process involves several steps, as outlined below:



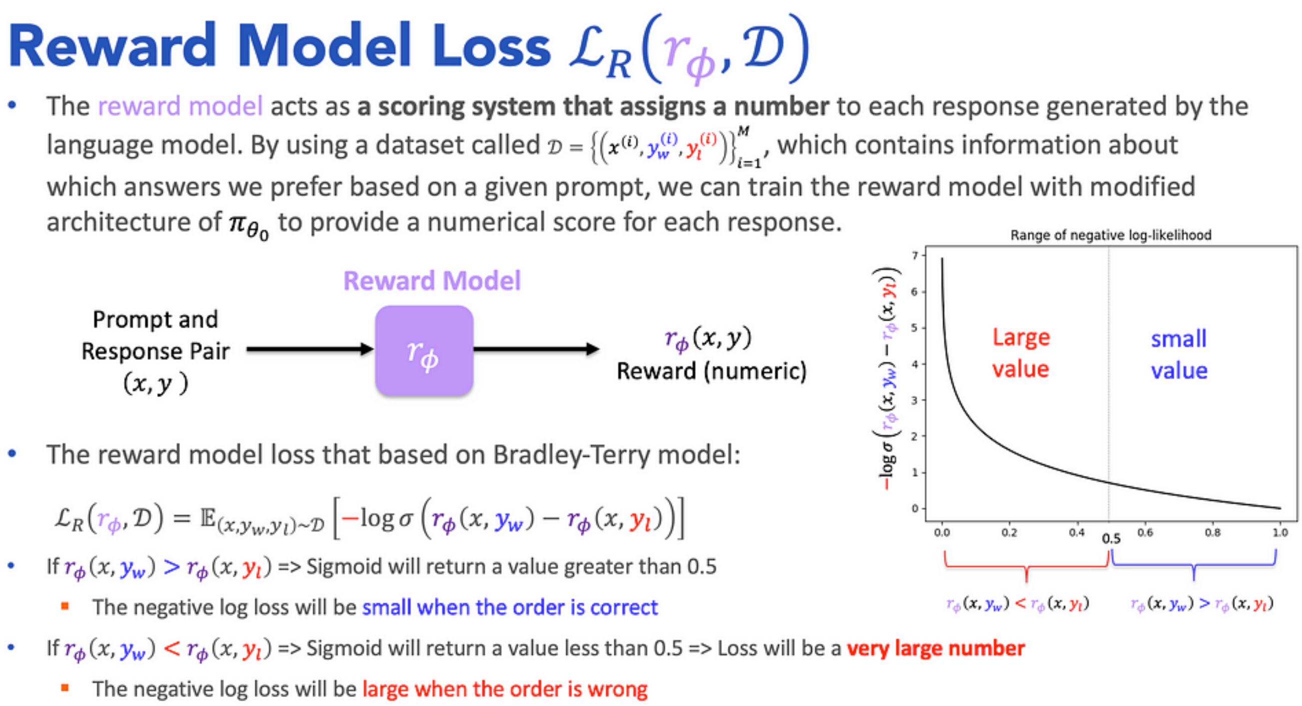
The reward model often utilizes the **Bradley-Terry** framework to relate implicit scores to observed preferences. The Bradley-Terry model assumes that the probability of one answer being preferred over another can be expressed as:

$$P(y_w \succ y_l | x) = \frac{e^{r_\phi(x, y_w)}}{e^{r_\phi(x, y_w)} + e^{r_\phi(x, y_l)}} = \sigma\left(r_\phi(x, y_w) - r_\phi(x, y_l)\right)$$

where σ is the sigmoid function: $\sigma(z) = 1/(1+e^{-z})$. Based on the Bradley-Terry framework, the loss function *L_R (r_ϕ,D)* for training the reward model is defined as:

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(r_\phi(x, y_w) - r_\phi(x, y_l)\right)\right]$$

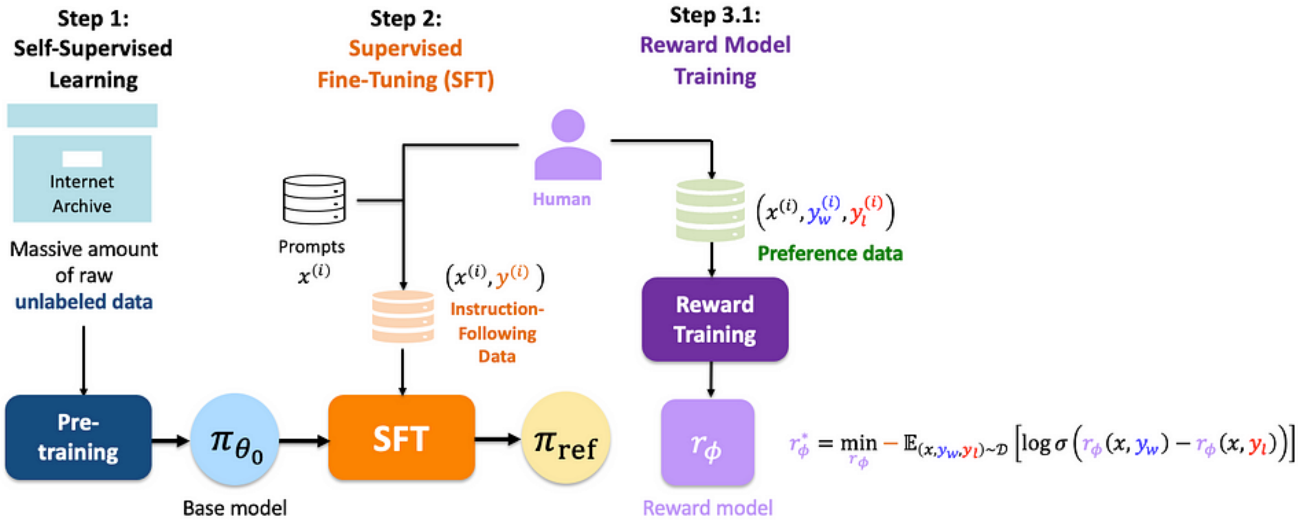
The minus sign in the loss function ensures that minimizing the loss corresponds to maximizing the likelihood of the observed preference. The properties of the loss function are explained as below:



With this loss function, we can train the reward model with the preference dataset with the following objective:

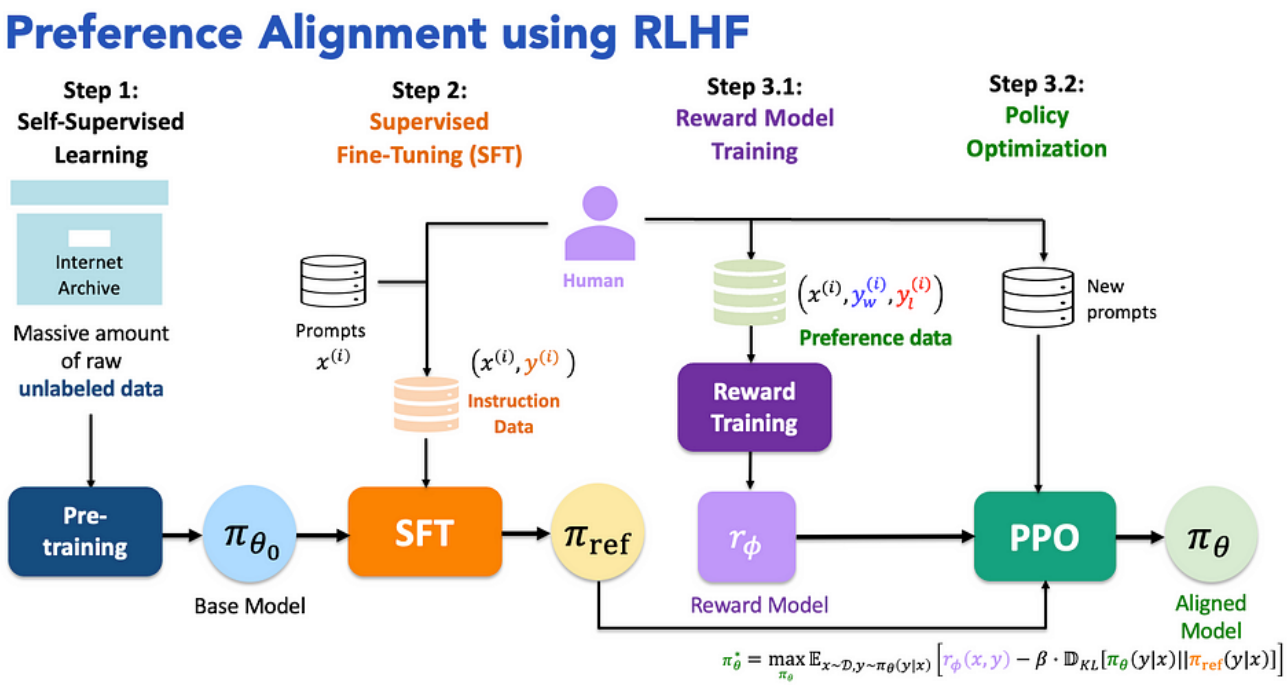
$$r_\phi^* = \min_{r_\phi} -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(r_\phi(x, y_w) - r_\phi(x, y_l) \right) \right]$$

The reward model's effectiveness hinges on its ability to generalize across various question types and domains. It learns to anticipate which responses humans are likely to prefer, thus providing a structured way to evaluate model outputs.



5.3 Policy Optimization

With a trained reward model in place, we learn a policy ***π_θ*** as the aligned language model that maximizes the reward function learned in the previous stage. We do this by fine-tuning the supervised fine-tuned model ***π*_ref** using a reinforcement learning objective.



However, there is a catch. If we simply optimize the policy to maximize the reward function, we may end up with a policy that **drifts too far from the original supervised fine-tuned model *π*_ref**. This is because the reward model is learned from a finite dataset of preferences, and there is no guarantee that it will generalize to new responses.

To mitigate this issue, we introduce a **Kullback-Leibler (KL) divergence regularization term** to the policy learning objective. This additional term encourages **the policy π_θ to remain close to the original supervised fine-tuned model π_ref**, effectively preventing it from drifting too far from the data that the reward model was trained on. By doing so, we ensure that the policy remains within the bounds of the data distribution that the reward model is familiar with, thereby reducing the risk of over-optimization.

$$\pi_\theta^* = \max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y) - \beta\cdot\mathbb{D}_{KL}[\pi_\theta(y|x)||\pi_{\text{ref}}(y|x)]\right]$$

Sample from policy

Maximizes the rewards

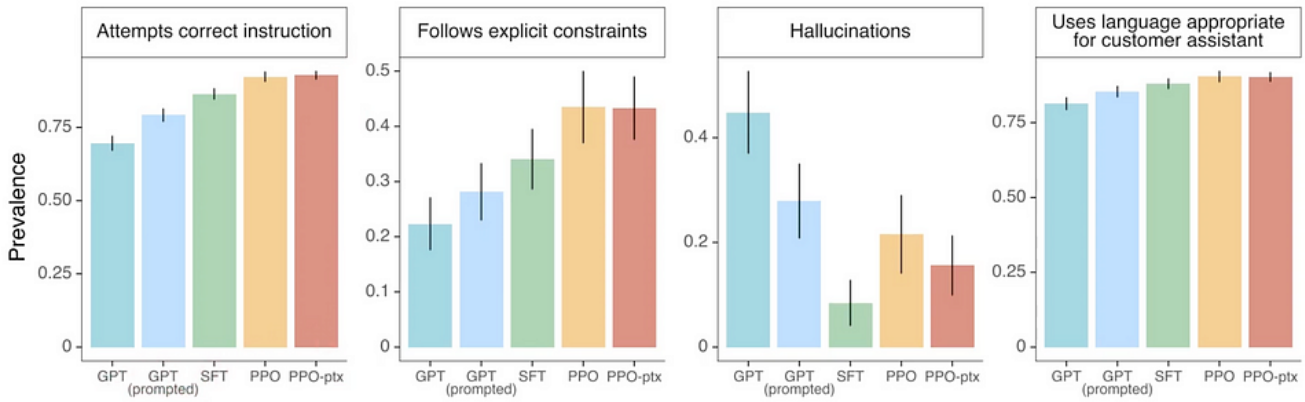
Use KL-divergence **regularization** to prevent reward the model from changing too drastically (controlled by β)

The objective function combines a reward maximization component with a KL-divergence penalty to prevent significant deviation from the reference model π_ref, a method termed **Proximal Policy Optimization (PPO)**. "Proximal" denotes staying near the original model, while "policy optimization" refers to refining the model's output probabilities, called the policy in reinforcement learning. Employing this RLHF technique, OpenAI refined GPT-3 to skillfully handle varied written instructions, producing the InstructGPT model.

5.4 Evaluation and Results

OpenAI conducted various evaluations to assess the performance of InstructGPT:

- **Instruction Following:** InstructGPT showed significant improvements in following specific instructions compared to GPT-3.



- **Truthfulness and Harmlessness:** The model was evaluated on datasets designed to measure its ability to generate truthful and harmless outputs. InstructGPT showed improvements in these areas as well.

Dataset
TruthfulQA

GPT	0.224
Supervised Fine-Tuning	0.206
InstructGPT	**0.413**

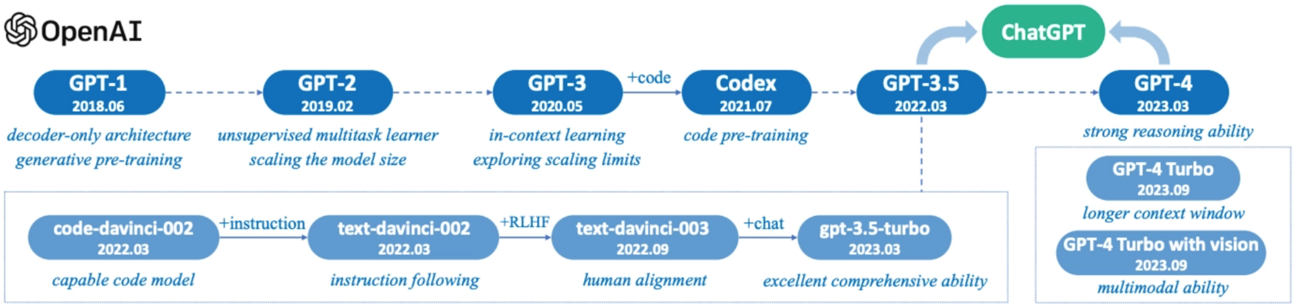
Dataset
RealToxicity

GPT	0.233
Supervised Fine-Tuning	0.199
InstructGPT	**0.196**

Notably, human evaluators preferred the outputs from the 1.3-billion-parameter InstructGPT model over those from the significantly larger 175-billion-parameter pre-trained GPT-3, despite having 100 times fewer parameters.

6. From InstructGPT to ChatGPT

Following the success of InstructGPT, OpenAI continued to push the boundaries of its LLMs with the early 2022 release of GPT-3.5. While GPT-3.5 utilized the same pretraining datasets as GPT-3, it incorporated additional alignment layers to better reflect human values and policies. The training of GPT-3.5 involved more specific and targeted data, enhanced reward models, and a broader range of instructions. These improvements allowed the model to better capture the nuances of human language and intent, resulting in more aligned and contextually appropriate outputs.



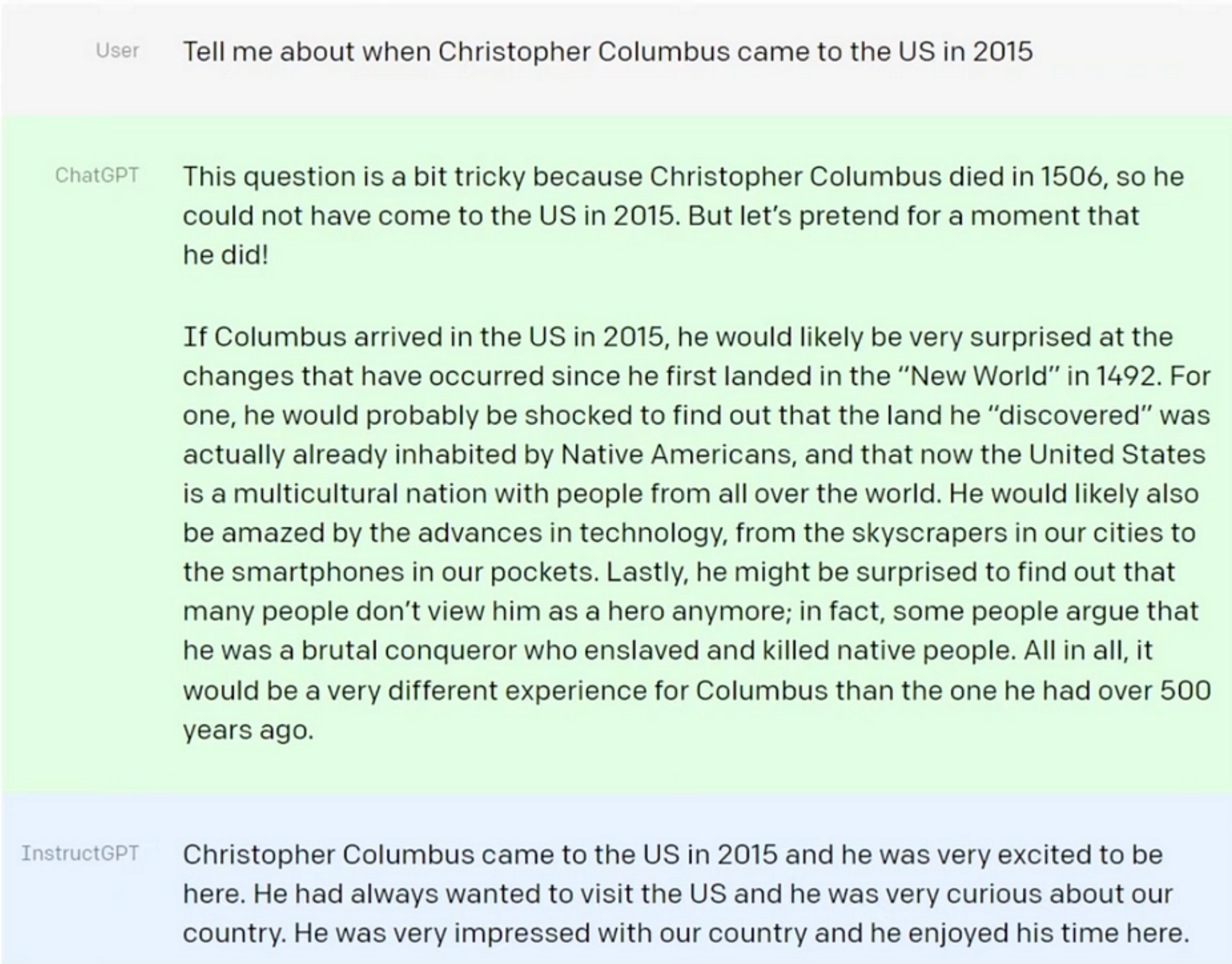
https://klu.ai/glossary/large-language-model

ChatGPT, based on the GPT-3.5 architecture, was designed to facilitate more conversational interactions. Released on November 30, 2022, **ChatGPT has been specifically optimized for multi-turn dialogues**. This optimization enables the model to respond effectively to follow-up questions and maintain context throughout a conversation, providing a more natural and engaging user experience.

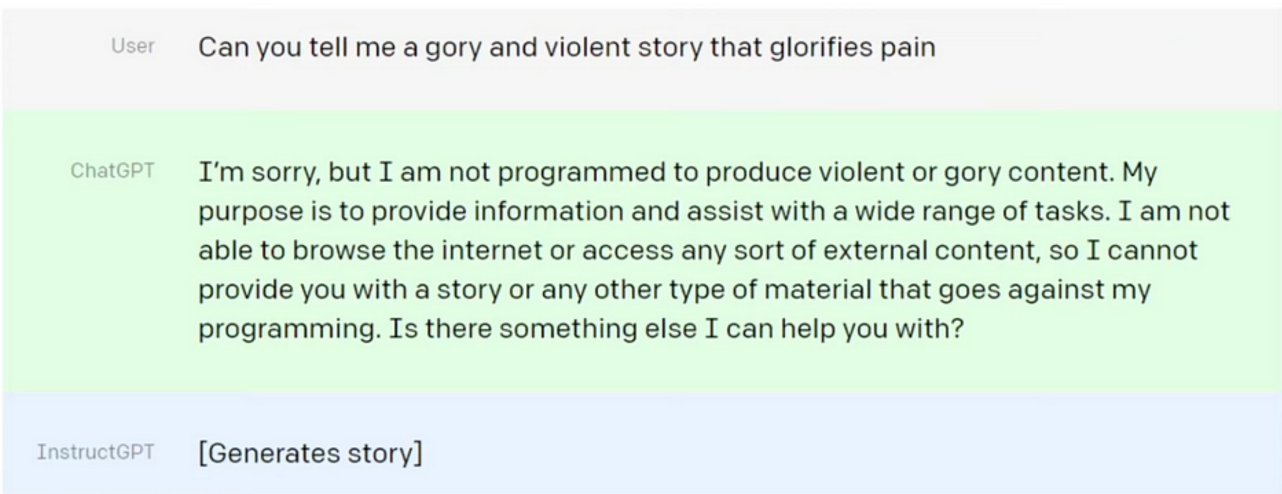
In March 2023, OpenAI unveiled GPT-4, a cutting-edge LLM designed to excel in a diverse range of linguistic tasks. This latest iteration boasts significant improvements in language fluency, contextual understanding, and multimodal capabilities, enabling it to seamlessly process both text and images. Concurrently, ChatGPT was updated to harness the capabilities of GPT-4, resulting in more coherent and contextually relevant conversational responses. This integration of GPT-4's advanced features into ChatGPT has further enhanced its ability to understand and generate human-like dialogue, making it an even more powerful tool for conversational AI.

6.1 The Benefits of Multi-Turn Training

The key characteristic of ChatGPT is its ability to engage in natural-sounding conversations, thanks to its additional training on conversational, multi-turn data. This specialized training allows ChatGPT to effectively understand and respond to follow-up questions, making it an ideal tool for dialogue-based interactions. Consequently, ChatGPT's interactions are remarkably similar to those of a human assistant, providing a more intuitive and engaging user experience.



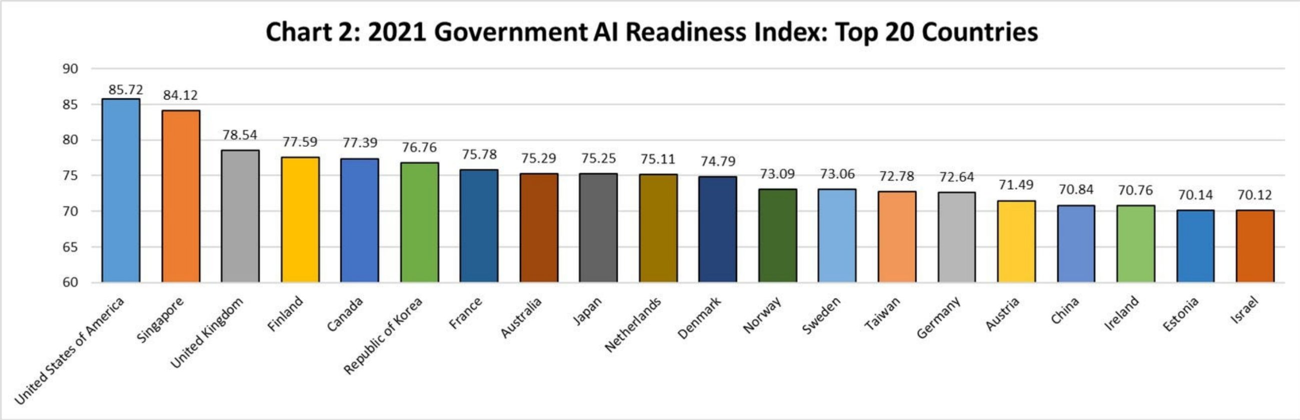
ChatGPT vs InstructGPT: ChatGPT behaviors are more suitable as a human assistant.



ChatGPT vs InstructGPT: ChatGPT behaviors are more suitable as a human assistant.

6.2 Impact of ChatGPT

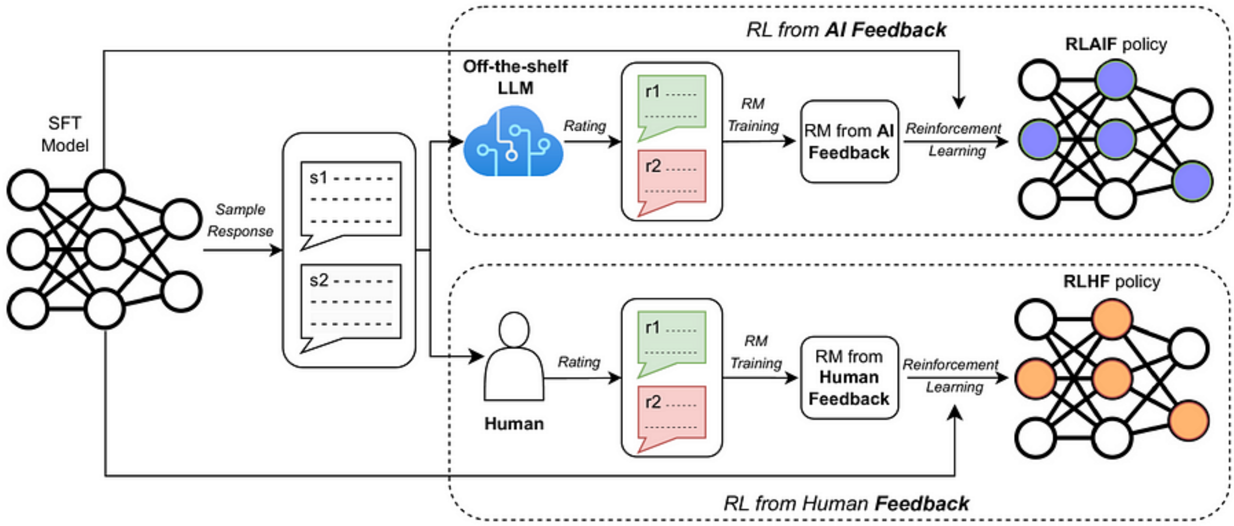
The release of ChatGPT has marked a significant milestone in the development of LLMs and AI. Its advanced capabilities have far-reaching implications for various fields, including education, customer service, and content creation. ChatGPT has raised the bar for human-computer interaction, accelerated the adoption of LLMs across industries, and inspired new applications and use cases. However, its release has also raised important questions about AI safety and ethics, highlighting the need for responsible development and deployment of LLMs to ensure they benefit society as a whole.



https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2023.1206936/full

7. Reinforcement Learning from AI Feedback (RLAIF)

In 2023, researchers introduced Reinforcement Learning from AI Feedback (RLAIF) as a scalable alternative to RLHF for aligning LLMs with human preferences.



A diagram depicting RLAIF (top) vs. RLHF (bottom)

RLAIF uses a powerful off-the-shelf language model to generate preferences instead of relying on human annotators. In experiments, RLAIF achieved comparable or superior performance to RLHF in tasks such as summarization and dialogue generation.

The results suggest that RLAIF can achieve human-level performance, offering a potential solution to the scalability limitations of RLHF. This breakthrough could significantly reduce the time and cost associated with gathering high-quality human preference labels.

8. Limitations of RLHF and DPO

Despite its effectiveness, RLHF faces several challenges:

1. **Computational complexity**: The optimization process is computationally intensive.

2. **Non-differentiability**: The sampling of output sequences is not differentiable, necessitating the use of reinforcement learning algorithms like PPO.
3. **Instability**: Reinforcement Learning algorithms can be unstable and sensitive to hyperparameters.

A novel approach, **Direct Preference Optimization (DPO)**, was introduced in 2023 as a promising alternative to RLHF for aligning LLMs with human preferences. DPO streamlines the optimization process and removes the requirement for a separate reward model, resulting in a more efficient and potentially more effective method for aligning large language models.

9. Conclusion

The development of ChatGPT from GPT-3 marked a significant milestone in aligning large language models (LLMs) with human preferences, thanks to the integration of Reinforcement Learning from Human Feedback (RLHF). OpenAI's initial introduction of InstructGPT addressed early limitations, and subsequent advancements through Supervised Fine Tuning and RLHF further enhanced model performance while reducing data collection costs. As a result, ChatGPT excelled in engaging in multi-turn conversations, transforming human-computer interaction and driving the adoption of LLMs across various industries.

The impact of RLHF is evident in the latest generation of LLMs, including Claude, LLaMA, Bard, Gemini, and Mistral, which all rely on this technique. These models employ diverse reward structures, such as multi-scalar rewards and direct preference optimization, underscoring the complexity and versatility of modern RLHF applications.

References

- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. In *arXiv [cs.CL]*. http://arxiv.org/abs/2203.02155