Reinforcement Learning with Human Feedback

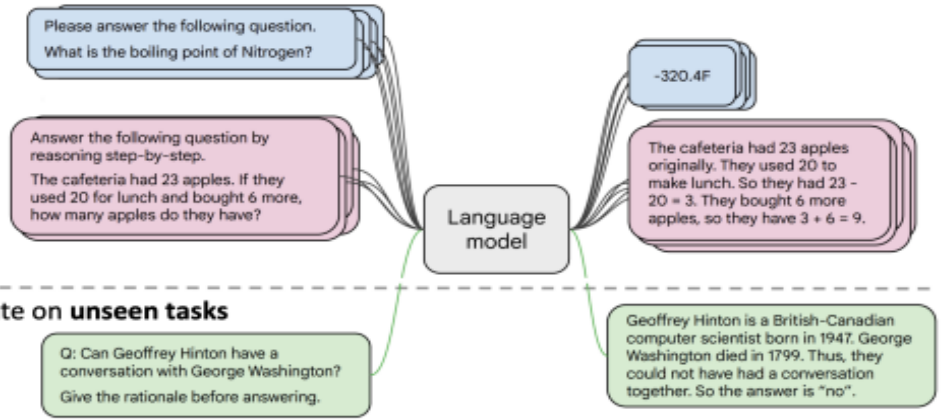
1 Why Reinforcement Learning for Human Feedback?

1.1 Limitations of Instruction Finetuning

Instruction finetuning takes an existing model and fine-tunes it using example pairs of natural language instructions and output. Examples of pairs can be found in the graphic below. These examples help the model understand specifically how to behave when given instructions. As a result, the model is then able to evaluate unseen tasks in a similar way. Before instruction tuning, these models do not know how to behave with many types of natural language instructions and would fail to solve the task.

Instruction finetuning

- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM



[FLAN-T5; Chung et al., 2022]

While instruction tuning is simple and straightforward and helps a model to generalize to unseen tasks, is has multiple limitations. One limitation is that it can be difficult and expensive to develop or collect a comprehensive set of rules for every possible combination of instructions. This means that some optimizations may be missed or not fully exploited. Additionally, the optimization process can be computationally expensive, as it involves exploring a large search space of possible instruction sequences. Another limitation of instruction finetuning is that many, often creative tasks, do not have a right answer and as such it is diffcult or impossible to provide adequate examples. Finally, these models penalizes all token-level mistakes equally even though some mistakes may be worse than others. As such there is a mismatch between the LM objective and the objective to satisfy human preferences.

What is Reinforcement Learning from Human Feedback (RLHF)?

In traditional RL, the reward signal is defined by a mathematical function based on the goal of the task at hand. However, in some cases, it may be difficult to specify a reward function that captures all the aspects of a task that are important to humans. For example, in the case of a robot that cooks pizza, the automated reward system may be able to measure objective factors such as crust thickness and amount of sauce and cheese, but it may not be able to capture the subjective factors that make a pizza delicious.

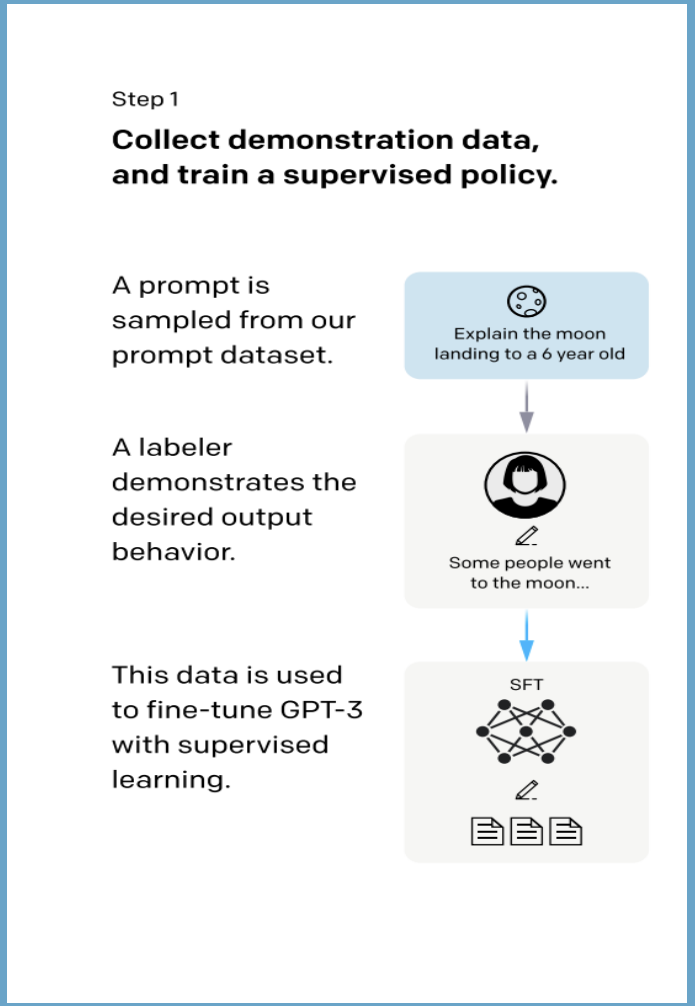
This is where Reinforcement learning from human feedback (RLHF) comes in. RLHF is a method of training RL agents that incorporates feedback from human supervisors to supplement the automated reward signal. By doing so, the RL agent can learn to account for the aspects of the task that the automated reward function cannot capture.

However, relying solely on human feedback is not always practical because it can be time-consuming and expensive. Therefore, most RLHF systems use a combination of automated and human-provided reward signals. The automated reward system provides the primary feedback to the RL agent, and the human supervisor provides additional feedback to supplement the automated reward signal. This may involve providing occasional rewards or punishments to the agent or providing data to train a reward model that can help improve the automated reward signal.

One advantage of RLHF is that it can help improve the safety and reliability of RL agents by allowing humans to intervene and provide feedback when the agent is performing poorly or making mistakes. Additionally, RLHF can help ensure that the agent is learning to perform the task in a way consistent with human preferences and values.

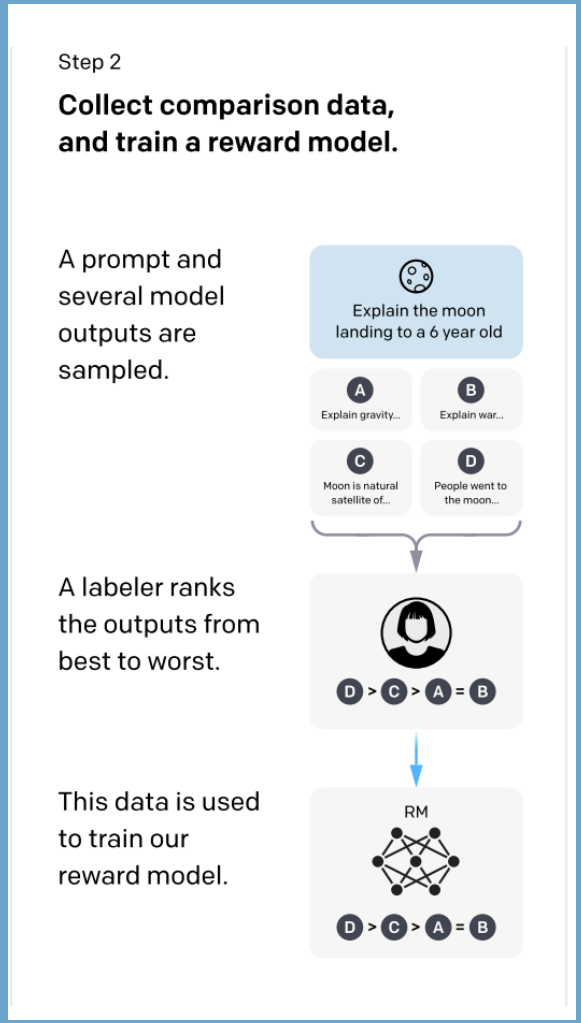
Training language models to follow instructions with human feedback

- Step 1:
  - Collect demonstration data from users .
  - The labelers provide demonstrations of the desired behavior on the input prompt distribution.
  - Then, a pretrained GPT-3 model is fine-tuned on this data using supervised learning



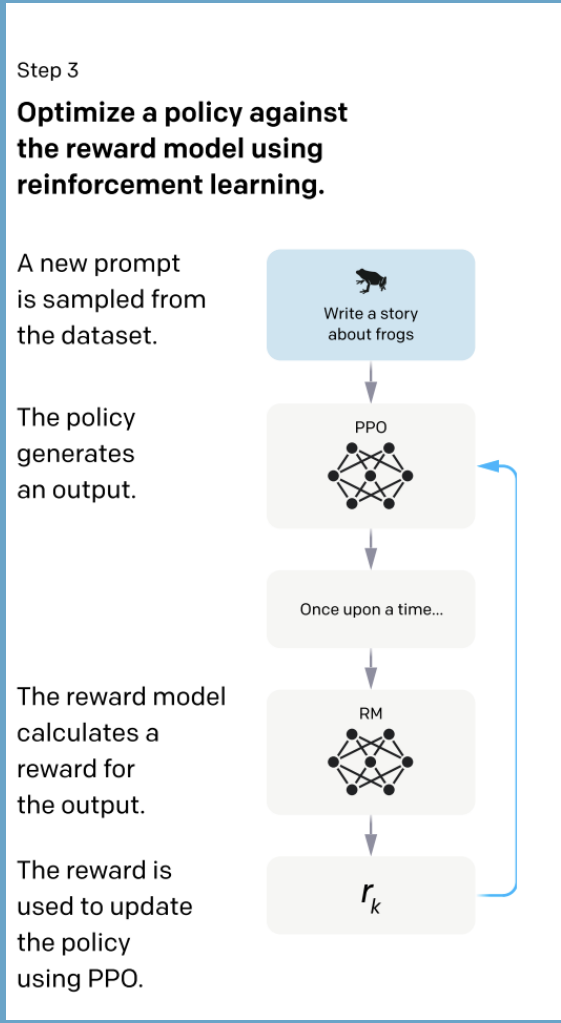
Training language models to follow instructions with human feedback

- Step 2:
  - Collect comparison data, and train a reward model.
  - A dataset of comparisons between model outputs is collected, where labelers indicate which output they prefer for a given input.
  - Then, a reward model is trained to predict the human-preferred output



Training language models to follow instructions with human feedback

- Step 3:
  - A policy is optimized against the reward model using PPO.
  - The output of the RM is used as a scalar reward.
  - The supervised policy is fine-tuned to optimize this reward using the PPO algorithm

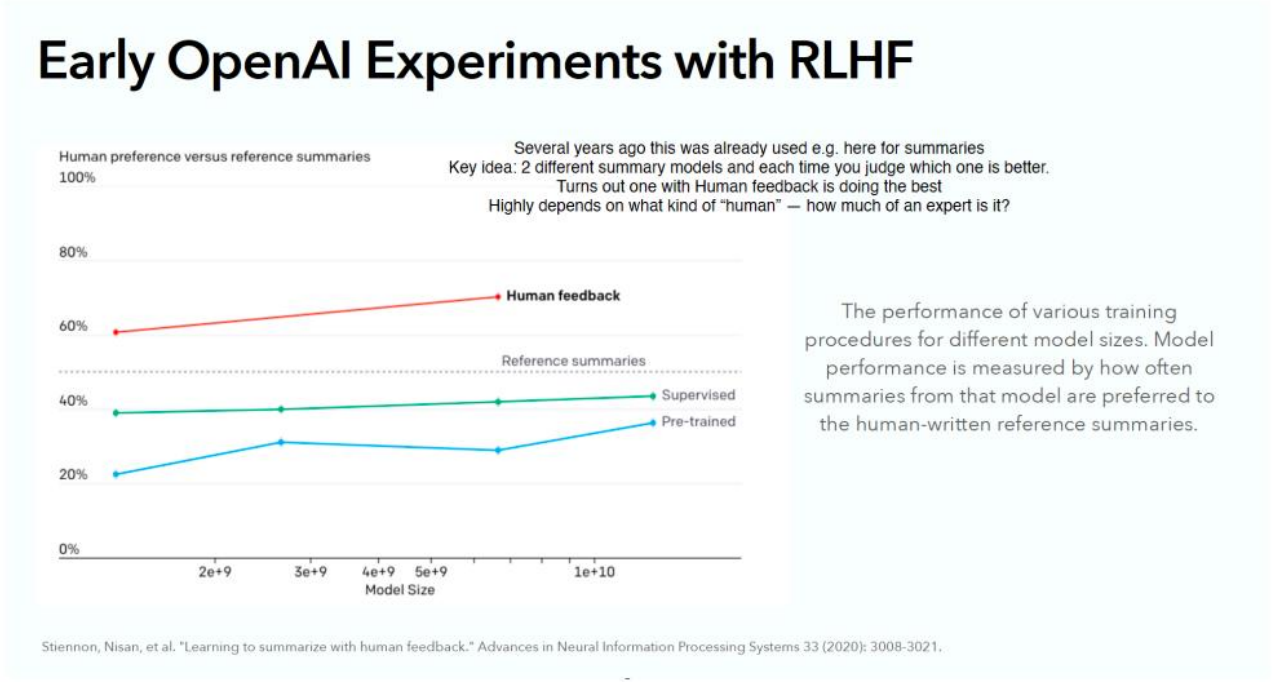


Some applications of RLHF

- **Game playing:** Human feedback can play a vital role in improving the performance of AI agents in game-playing scenarios. With feedback from human experts, agents can learn effective strategies and tactics that work in different game scenarios. For example, human feedback can help an AI agent improve its gameplay and decision-making skills in the game of Go.
- **Personalized recommendation systems:** Personalized recommendation systems rely on human feedback to learn the preferences of individual users. By analyzing user feedback on recommended products, the agent can learn which features are most important to them. This allows the agent to provide more personalized recommendations in the future, improving the overall user experience.
- **Robotics human feedback:** Robotics human feedback is crucial in teaching AI agents how to interact with the physical environment safely and efficiently. In robotics, an AI agent could learn to navigate a new environment more quickly with feedback from a human operator on the best path to take or which objects to avoid. This can help improve the safety and efficiency of robots in various applications, such as manufacturing and logistics.
- **Education AI-based tutors:** Education AI-based tutors can use human feedback to personalize the learning experience for students. With feedback from teachers on which teaching strategies work best with different students, an AI-based tutor can help students learn more effectively. This can lead to improved learning outcomes and a better student learning experience.

2.3 History of RLHF

While RLHF has only recently become well known, it has indeed a long history. In earlier days (around 2008), it was often used in robotics. OpenAI also employed this method years ago and showed that Human feedback models outperform supervised or pretrained models. The graphic below shows the results in detail.

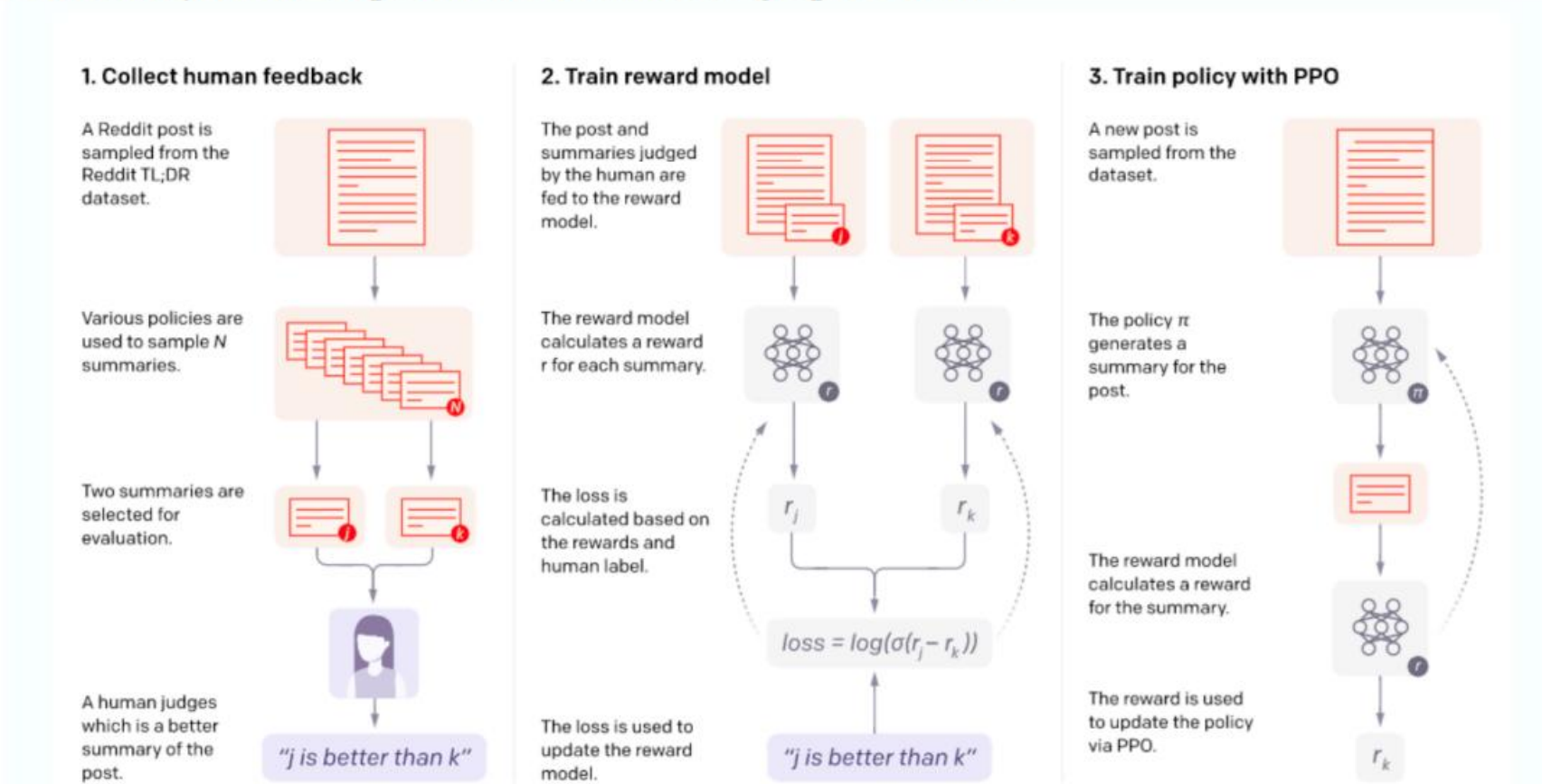


2.4 How did RLHF work?

Basic RLHF (before modern versions) worked in 3 steps. First, human feedback is collected. For example, a human would judge which of two summaries of a reddit post that the algorithm generated are better. Second, a reward model is trained based on the model output and human feedback pairs. Within that reward model, a loss function calculates the loss based on the rewards and the human label. In a third step, the policy is trained with PPO (Proximal Policy Optimization). During that step, a new sample input is used and the policy generated the desired output (e.g. a summary). The reward model caculates the reward and this reward is then used to update the policy again via PPO.

Continued to Next Slide

A summary of the steps can be found in the graphic below.

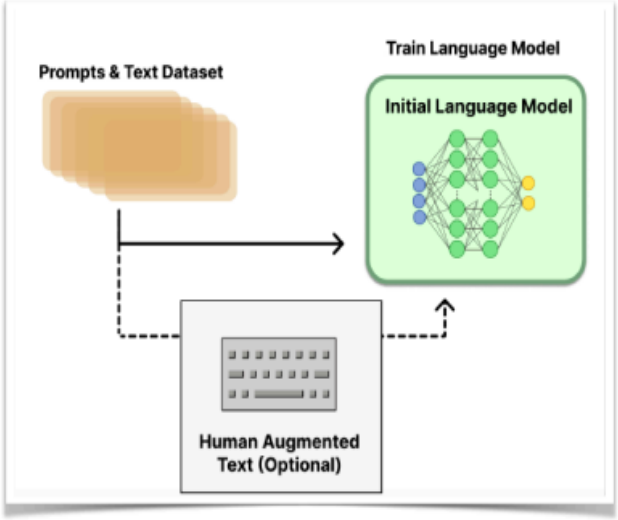


experts judge

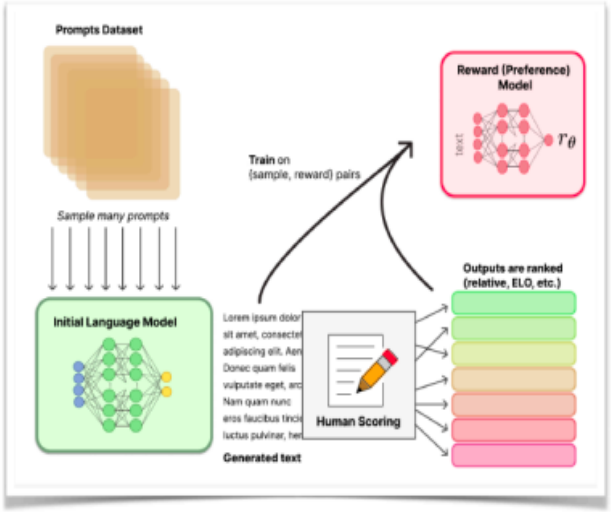
model learns what is better

Modern RLHF Overview

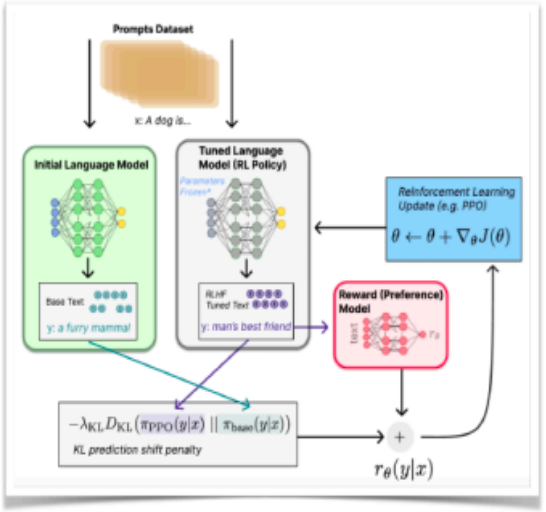
1. Language model pretraining



2. Reward model training



3. Fine-tuning with RL



2.5 How does modern RLHF work?

Modern RLHF works slightly different, but also involves 3 steps: 1. Language model pretraining, 2. Reward Model Training, 3. Fine-Tuning with RL.

In 1. Language Model pretraining, usually a lot of data is collected from the web (e.g. reddit - good data base for prompts and answers). This is called Unsupervised sequence prediction. Optionally, human-written text from prompts are included as well - "Supervised fine tuning". This is usually expensive but is viewed as "high quality". After the data is collected, the initial language model is trained.

In 2. Reward Model Training, the key goal is to catch human preferences in modeling rewards. First, a prompt data set is inputed into the initial language model trained in step 1. Then, the generated text is scored by humans. Since this is expensive, we do not directly ask humans for preferences, but rather model them as a separate NLP problem. As such, we ask humans to rank their preferences by pairwise comparisons which is more reliable than asking for direct rankings. Based on this input, a reward model is trained.

In 3. Fine-Tuning with RL, the original model is fine-tuned using the reward function.

An overview can be found in the graphic below.

2.6 Human Feedback Interfaces

There are multiple ways to collect human feedback even after the model is deployed (e.g. ChatGPT). One option is to allow users to upvote/downvote the machine generated response. Another option is to give users multiple alternative reponses and let them choose the best one. Humans could also edit the output text in the interface and the model could learn what part of the output should be modified.

2.7 Limitations of RLHF

RLHF is a very powerful method, yet it has a few limitations. First, collecting human feedback at scale is extremely expensive since humans need to be paid. If humans are included, one must consider that the quality of the human feedback that can highly influence the model performance. Experts may judge information very differently than novices. Further, running such models (like OpenAI's ChatGPT) are computationally expensive and thus cost a lot of money. Finally, RLHF has another serious limitation: human preferences are unreliable and as such, "reward hacking" is a common problem. Models are rewarded responses that seem authoritative and helpful, regardless of truth. This can lead to models making up facts and hallucinations.

Benefits of RLHF

1. **Improved performance:** RLHF incorporates human feedback into the learning process, enabling AI systems to understand complex human preferences better and generate more accurate, coherent, and contextually relevant responses. This leads to enhanced performance and increased user satisfaction.
2. **Adaptability:** RLHF allows AI models to adapt to different tasks and scenarios by leveraging human trainers' diverse experiences and expertise. This flexibility enables the models to excel in various applications, including conversational AI and content generation.
3. **Reduced biases:** Through an iterative feedback process, RLHF helps identify and mitigate biases present in the initial training data. Human trainers evaluate and rank the model-generated outputs, ensuring alignment with human values and minimizing unwanted biases.
4. **Continuous improvement:** RLHF facilitates ongoing improvement in model performance. As trainers provide more feedback and the model undergoes reinforcement learning, it becomes increasingly proficient in generating high-quality outputs, resulting in continuous enhancements.

Benefits of RLHF

5. **Enhanced safety:** RLHF contributes to developing safer AI systems by allowing human trainers to guide the model away from generating harmful or undesirable content. This feedback loop ensures greater reliability and trustworthiness in AI interactions with users.
6. **User-centric design:** By incorporating human feedback, RLHF helps AI systems better understand user needs, preferences, and intentions. This leads to more personalized and engaging experiences as the models generate responses tailored to individual users.
7. **Efficient training:** RLHF improves the efficiency of training large language models by leveraging human feedback to guide the learning process effectively. This saves time and computational resources, making the training process more efficient.
8. **Leveraging domain expertise:** RLHF enables AI models to benefit from human trainers' expertise and domain knowledge. By collecting feedback from trainers with diverse backgrounds and perspectives, the models learn to generate responses that represent various viewpoints and address specific user needs.

How is RLHF used in large language models like ChatGPT?

LLMs have become a fundamental tool in Natural Language Processing (NLP) and have shown remarkable performance in various language tasks such as language modeling, machine translation, and question-answering. However, even with their impressive capabilities, LLMs still suffer from limitations, such as being prone to generating low-quality, irrelevant, or even offensive text.

One of the main challenges in training LLMs is obtaining high-quality training data, as LLMs require vast amounts of data to achieve high performance. Additionally, human annotators are needed to label the data for supervised learning, which is a time-consuming and expensive process.

To overcome these challenges, RLHF was introduced as a framework that can provide high-quality labels for training data. In this framework, the LLM is first pre-trained through unsupervised learning and then fine-tuned using RLHF to generate high-quality, relevant, coherent text.

RLHF allows LLMs to learn from human preferences and generate outputs more aligned with user goals and intents, which can have significant implications for various NLP applications. By combining reinforcement learning and human feedback, RLHF can efficiently train LLMs with less labeled data and improve their performance on specific tasks. Therefore, RLHF is a powerful framework for enhancing the capabilities of LLMs and improving their ability to understand and generate natural language.

In RLHF, the LLM is first pre-trained through unsupervised learning on a large corpus of text data. This allows the model to learn the underlying patterns and structures of language, essential for generating coherent and meaningful outputs. Pre-training the LLM is computationally expensive, but it provides a solid foundation that can be fine-tuned using RLHF.

The second phase involves creating a reward model, a machine learning model that evaluates the quality of the text generated by the LLM. The reward model takes the output of the LLM as input and produces a scalar value that represents the quality of the output. The reward model can be another LLM modified to output a single scalar value instead of a sequence of text tokens.

To train the reward model, a dataset of LLM-generated text is labeled for quality by human evaluators. The LLM is given a prompt, generating several outputs that human evaluators rank from best to worst. The reward model is then trained to predict the quality score of the LLM-generated text. The reward model creates a mathematical representation of human preferences by learning from the LLM's output and the ranking scores assigned by human evaluators.

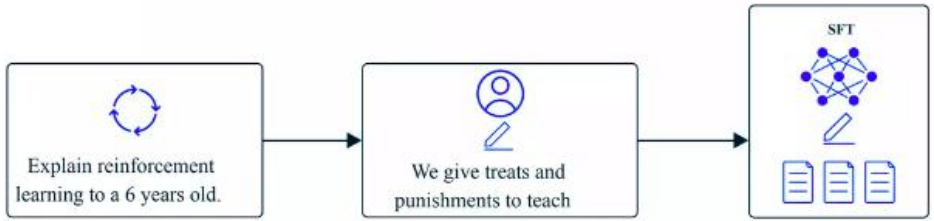
In the final phase, the LLM becomes the RL agent, creating a reinforcement learning loop. The LLM takes several prompts from a training dataset in each training episode and generates text. Its output is then passed to the reward model, which provides a score that evaluates its alignment with human preferences. The LLM is then updated to generate outputs that score higher on the reward model.

One of the challenges of RLHF is maintaining a balance between reward optimization and language consistency. The reward model is an imperfect approximation of human preferences, and the RL agent might find a shortcut to maximize rewards while violating grammatical or logical consistencies. To prevent this, the ML engineering team keeps a copy of the original LLM in the RL loop. The difference between the output of the original and RL-trained LLMs, also known as the Kullback-Leibler divergence, is integrated into the reward signal as a negative value to prevent the model from drifting too much from the original output.

How Does Chatgpt Work?

Step 1

Collect demonstration data and train a supervised policy.



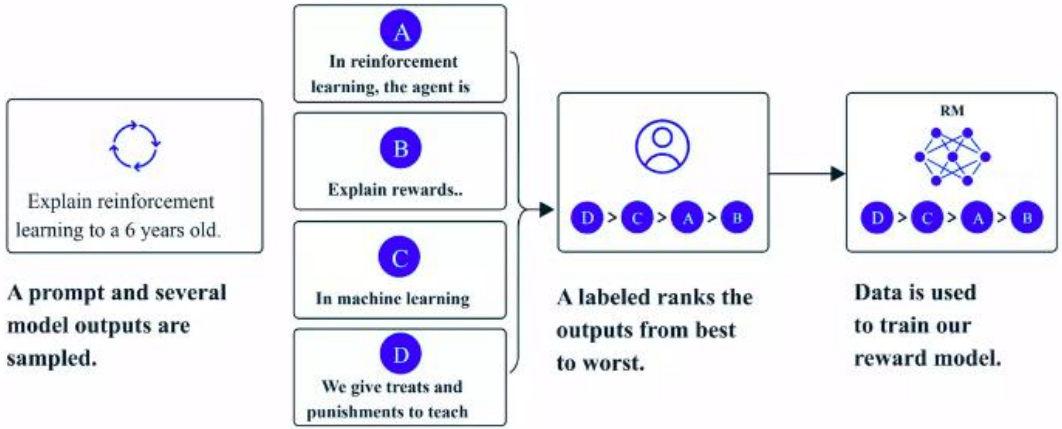
A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

Data is used to Fine-tune GPT-3.5 with supervised learning.

Step 2

Collect comparison data and train a reward model.



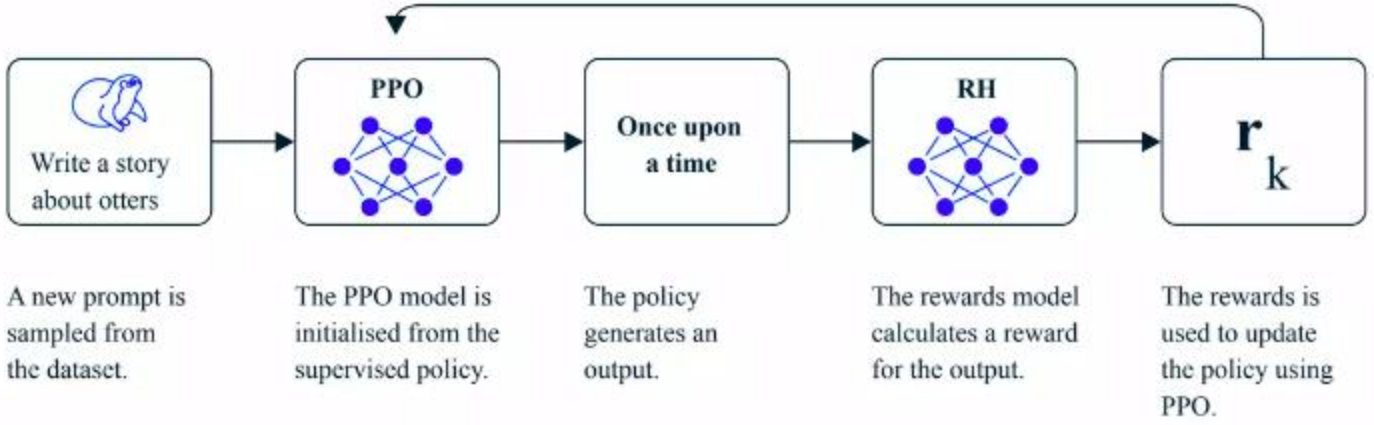
A prompt and several model outputs are sampled.

A labeled ranks the outputs from best to worst.

Data is used to train our reward model.

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm



ChatGPT, like other large language models, uses the RLHF framework to improve its performance on natural language generation tasks. However, some modifications to the general framework are specific to ChatGPT.

ChatGPT uses a “supervised fine-tuning” process in the first phase on a pre-trained GPT-3.5 model. This involves hiring human writers to generate answers to a set of prompts, which are then used to finetune the LLM. This process differs from unsupervised pre-training, the standard method for pre-training LLMs. Supervised fine-tuning allows ChatGPT to be customized for specific use cases and improves its performance on those specific tasks.

In the second phase, ChatGPT creates a reward model using the standard procedure of generating multiple answers to prompts and having them ranked by human annotators. The reward model is trained to predict the quality of the text generated by the main LLM. This allows ChatGPT to learn from human feedback and improve its ability to generate high-quality, relevant, and coherent text.

In the final phase, ChatGPT uses the proximal policy optimization (PPO) RL algorithm to train the main LLM. PPO is a popular RL algorithm used successfully in many applications, including natural language processing. ChatGPT takes several prompts from a training dataset in each training episode and generates text. The text is then evaluated by the reward model, which provides a score that evaluates its alignment with human preferences. The LLM is then updated to create outputs that score higher on the reward model.

To prevent the model from drifting too much from the original distribution, ChatGPT likely uses a technique called “KL divergence regularization”. KL divergence measures the difference between the output of the original and RL-trained LLMs. This difference is integrated into the reward signal as a negative value, which penalizes the model for deviating too far from the original output. Additionally, ChatGPT may freeze some parts of the model during RL training to reduce the computational cost of updating the main LLM.

These modifications allow ChatGPT to generate high-quality, relevant, and coherent text, making it one of the most advanced LLMs available today.

Endnote

Reinforcement learning from human feedback helps improve the accuracy and reliability of AI models. By incorporating human feedback, these models can learn to better align with human values and preferences, resulting in improved user experiences and increased trust in AI technology.

RLHF is particularly important in the case of generative AI models. Without human guidance and reinforcement, these models may produce unpredictable, inconsistent, or offensive outputs, leading to controversy and consequences that can undermine public trust in AI. However, when RLHF is used to train generative AI models, humans can help ensure that the models produce outputs aligned with human expectations, preferences and values.

One area where RLHF can have a particularly significant impact is chatbots and customer service in general. By training chatbots with RLHF, businesses can ensure that their AI-powered customer service is able to accurately understand and respond to customer inquiries and requests, resulting in a better overall user experience. Additionally, RLHF can be used to improve the accuracy and reliability of AI-generated images, text captions, financial trading decisions and even medical diagnoses, further highlighting its importance in developing and implementing AI technology.

As the field of AI continues to evolve and expand, we must prioritize the development and implementation of RLHF to ensure the long-term success and sustainability of generative AI as a whole.