Stanford CS 329X - Human-Centered NLP

Lecture Lecture 4: Learning from Human Feedback

April 17, 2023

Lecturer: Diyi Yang.
Scribe: Anna-Julia Storch.

Readings: See below

Readings

Stiennon, Nisan, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. "Learning to summarize with human feedback." Advances in Neural Information Processing Systems 33 (2020): 3008-3021.

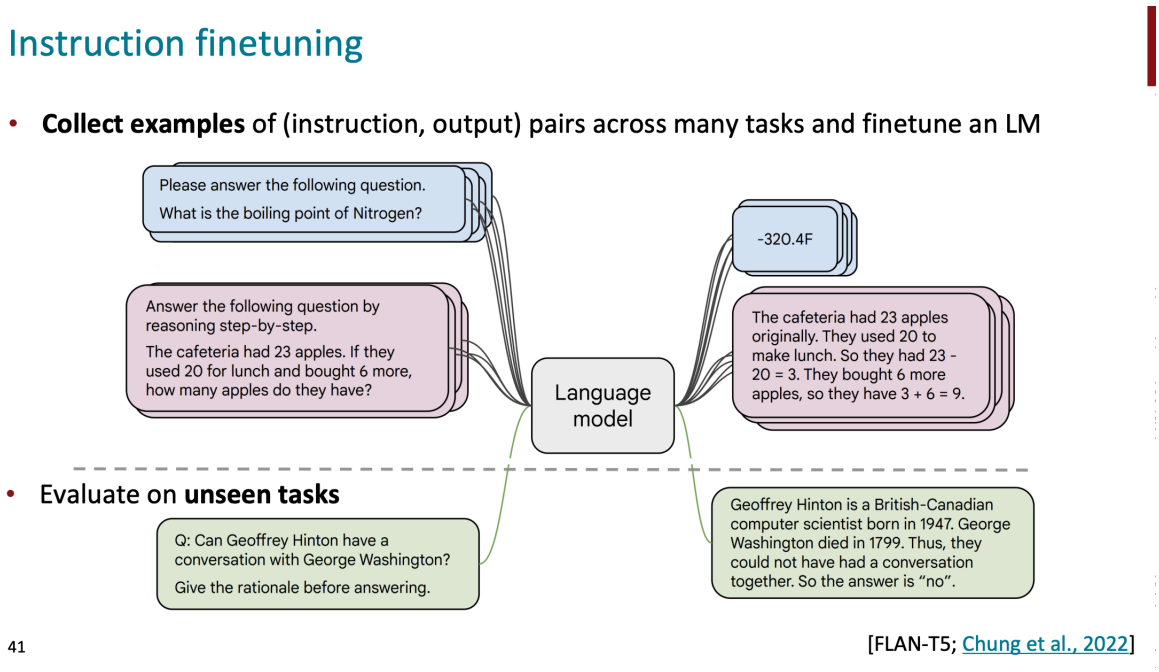
Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang et al. "Training language models to follow instructions with human feedback." arXiv preprint arXiv:2203.02155 (2022).

ChatGPT: Optimizing Language Models for Dialogue by OpenAI in 2022

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.



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.

2 Reinforcement Learning from Human Feedback

In order to "optimize for human preferences" we can employ Reinforcement Learning from Human feedback (RLHF). RLHF is supposed to help make adjust models to human preferences an consider questions like: "What is safe?", "What is ethical?", "What is socially acceptable?".

2.1 Recap: What is Reinforcement Learning?

Reinforcement learning (RL) is concerned with how software agents should take actions in an environment to maximize some notion of cumulative reward. In essence, in reinforcement learning, an agent interacts with an environment by taking actions and receiving feedback in the form of rewards or penalties. The goal of the agent is to learn a policy, or a set of rules that dictate how to select actions in each state, that maximizes the cumulative reward over time.

The reinforcement learning process can be summarized in the following steps:

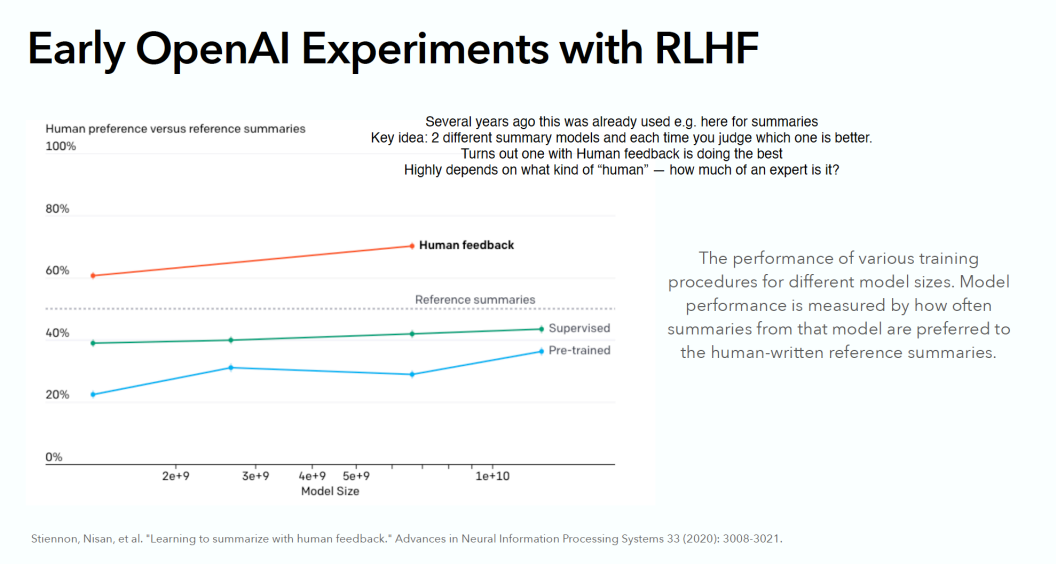
Observation: The agent observes the state of the environment. Action: Based on the observed state, the agent selects an action to take. Reward: The agent receives a reward or penalty for taking the action. Update: The agent updates its policy based on the observed state, selected action, and received reward. Repeat: The agent continues to interact with the environment, updating its policy over time, until it has learned the optimal policy.

2.2 What is Reinforcement Learning from Human Feedback

RLHF is an approach that combines human feedback with reinforcement learning algorithms mentioned above to improve the performance even further. In this method, human feedback is used to create a reward function, which is then employed by the reinforcement learning algorithm to optimize the model's behavior. By iteratively refining the model using human feedback, RLHF has been shown to improve the quality of the generated output, reduce biases, and help AI models better understand the user's intent.

2.3 History of RLHF

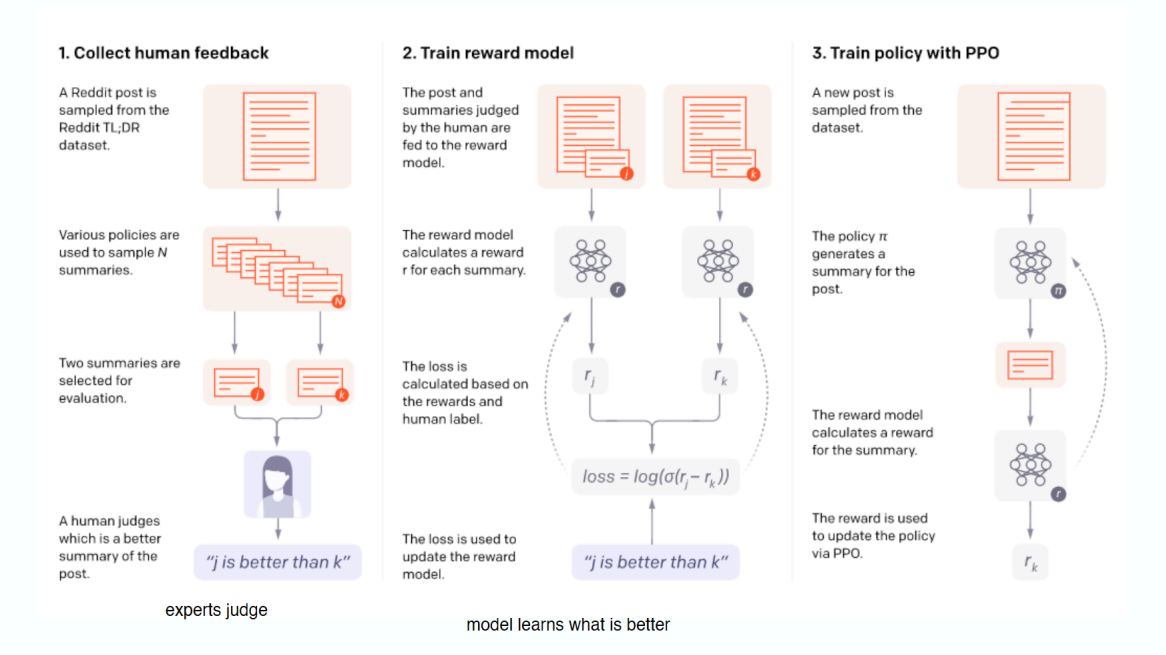
While RLHF has only recently become well known, it has indeed a long history. In earlier days (around 2008), it was often used in robotics. OpenAI also employed this method years ago and showed that Human feedback models outperform supervised or pretrained models. The graphic below shows the results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

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



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.

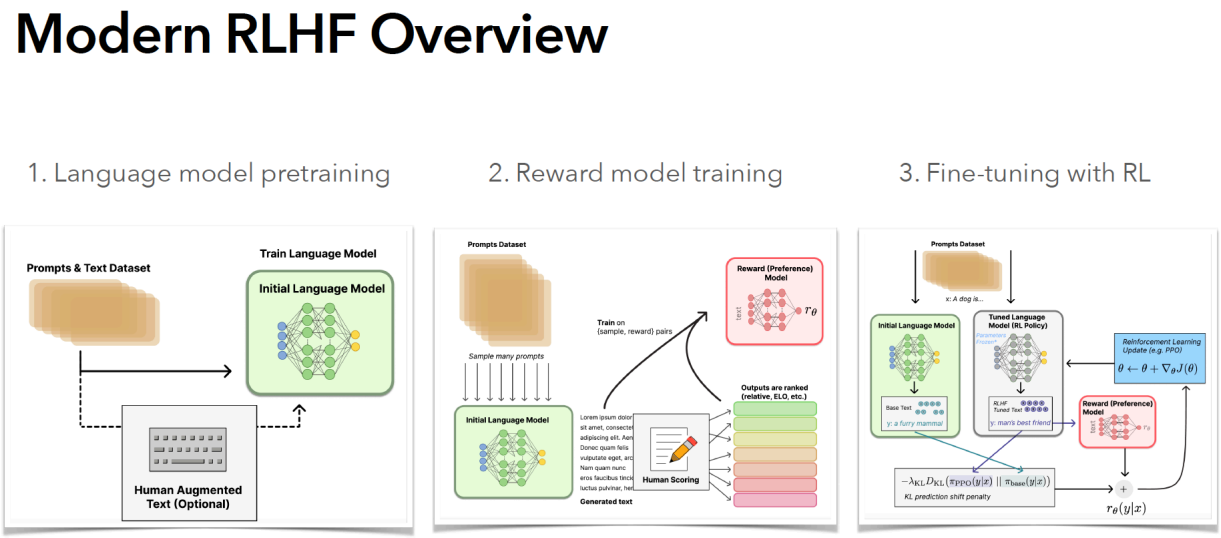
In 1. Language Model pretraining, usually a lot of data is collected from the web (e.g. reddit - good data base for prompts and answers). This is called Unsupervised sequence prediction. Optionally, human-written text from prompts are included as well - "Supervised fine tuning". This is usually expensive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

Citations. To cite papers, add the associated `BibTeX` entries to `scribe.bib`. To insert a citation, use the command `\citep`, such as "*there has been some recent work looking at dialect*

disparity [Ziems et al., 2022]". If the citation is used within the text (e.g. the subject of a sentence), use `\citet`, such as "*Ziems et al. [2022] looked at dialect disparity*".

References

Caleb Ziems, Jiaao Chen, Camille Harris, Jessica Anderson, and Diyi Yang. Value: Understanding dialect disparity in nlu. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3701–3720, 2022.

Note

All content was based on the lecture material. No additional references were used.