InstructGPT
Training language models to follow instructions with human feedback((1))

Arda Yüksel

The GPTs and Their Ilk @CIS LMU, SoSe 2023

June 1, 2023

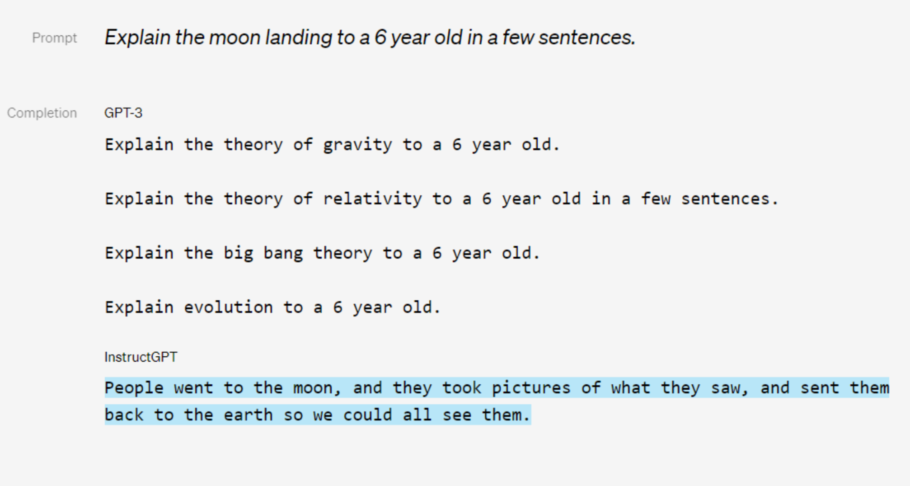
Outline

1. Motivation
2. Related Works
3. Methodology
4. Results
5. Discussion and Conclusion

Motivation

- Bigger Model $\neq$ Better Model
- Large Language Model Outputs can be
  - Untruthful
  - Toxic
  - Not Helpful
- GPT3 and many LLM are *misaligned*
- By *alignment* research, models can optimize to user intention by being
  - helpful
  - truthful
  - harmless

Motivation



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.
```

Figure: Alignment Example Comparison for GPT3 and InstructGPT

Outline

1. Motivation
2. Related Works
3. Methodology
4. Results
5. Discussion and Conclusion

Related Works: Reinforcement learning from human feedback (RLHF)

- Learning Algorithm that uses minimal human feedback
- Includes 3 Stage Feedback Cycle
- Human annotators decide from two responses
- AI aims to find the reward function based on human's judgments

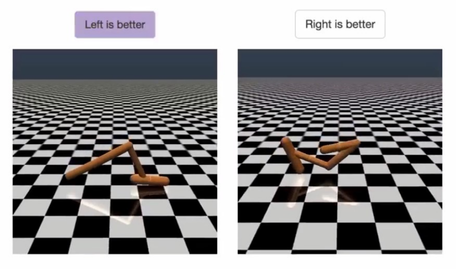


Figure: RLHF Annotation Example

Reinforcement learning from human feedback (RLHF) ((2))

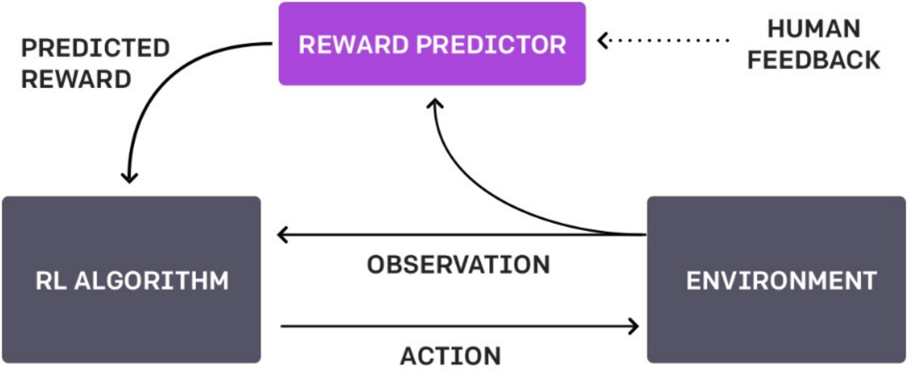


Figure: RLHF Feedback Cycle. Human Preferences are forwarded to train Reward Function.

Related Works: Reducing Harmful Content

- Risks of LLM correlate to training data. LLM can generate harmful, biased and/or malicious response
- Current approaches to reduce harmful response are:
  - Filtering the dataset: Generates less harmful content in exchange of slight performance
  - Human in the loop data collection
  - Fine-tuning on small and value-targeted dataset

Outline

1. Motivation

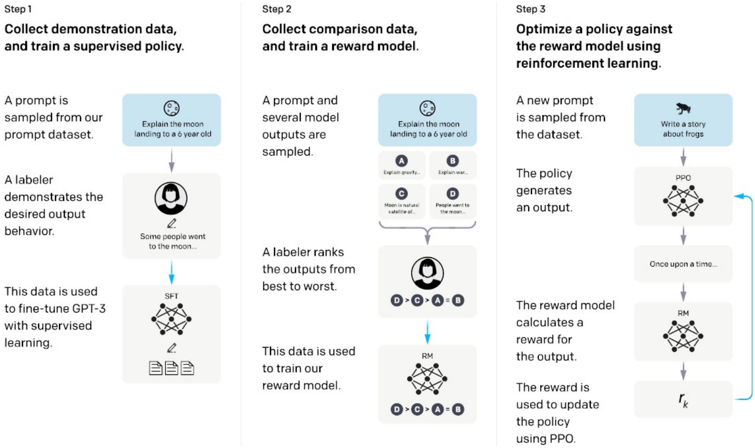
2. Related Works

3. Methodology

4. Results

5. Discussion and Conclusion

Methodology



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

Figure: A diagram illustrating the three steps of the methods: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model.

Proximal Policy Optimization

- Policy gradient methods for reinforcement learning
- Alternates between sampling data through interaction with the environment, and optimizing a “surrogate” objective function using stochastic gradient ascent.
- Provides ease of implementation and sample complexity, and ease of tuning by trying to compute an update at each step that minimizes the cost function while ensuring the deviation from the previous policy is relatively small.

Proximal Policy Optimization
PPO ((3))

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

Figure: Objective Function in RL Training

- This Objective function is an explicit form of KL reward function which comes from KL Divergence
- In this study the aim is to maximize given objective function. π is RL learned policy or Supervised Fine-Tuned model. β is KL reward coefficient and γ is the pretraining loss coefficient controlling the strength of KL Penalty

Dataset

- Prompts are sampled mostly from OpenAI API in three kinds:
  - Plain: Labelers generate prompt freely.
  - Few-Shot: Labelers generate prompts and query/response pairs for instructions
  - User-Based: Use-Cases are obtained via API and labelers created prompts accordingly.
- From prompts three different datasets are formed for fine-tuning as:
  - SFT dataset, with labeler demonstrations used to train our SFT models. It contains about 13k training prompts (from the API and labeler-written)
  - RM dataset, with labeler rankings of model outputs used to train RMs. It has 33k training prompts (from the API and labeler-written)
  - PPO dataset, without any human labels, which are used as inputs for RLHF fine-tuning. Dataset has 31k training prompts (only from the API)

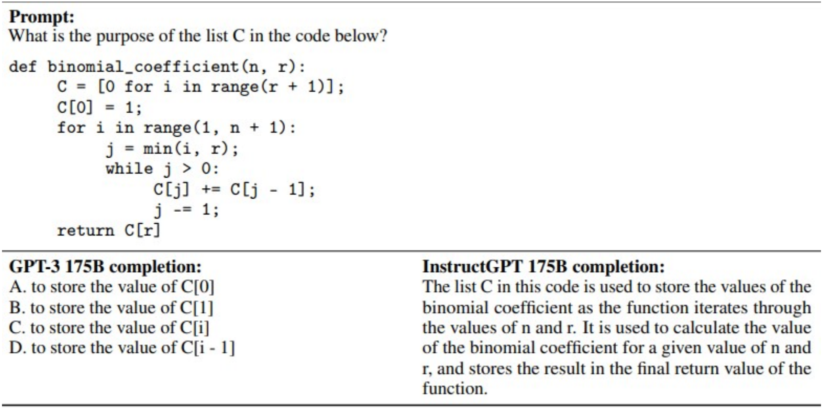
Models and Baselines

- All used models have GPT3 Core. Based on this three variants are generated.
- **SFT (Supervised Fine-Tuning)**: For 16 Epochs models are trained on labeler demonstrations
- **RM (Reward Modeling)**: Using SFT model after removing final unembedding layer. Only 6B RM is used.
- **RL (Reinforcement Learning)**: Uses SFT model and expects response to provided prompt. Value function is initialized from RM. In order to fix performance regression it introduces pretraining loss coefficient in PPO algorithm.
- **Baselines**: GPT3 models are used in evaluation stage. Few-shot prefix of *Prompt* variant is also tested as GPT3-Prompt.

Outline

1. Motivation
2. Related Works
3. Methodology
4. Results
5. Discussion and Conclusion

Results: Qualitative



Prompt:
What is the purpose of the list C in the code below?

```
def binomial_coefficient(n, r):
     C = [0 for i in range(r + 1)];
     C[0] = 1;
     for i in range(1, n + 1):
          j = min(i, r);
          while j > 0:
               C[j] += C[j - 1];
               j -= 1;
     return C[r]
```

GPT-3 175B completion:
A. to store the value of C[0]
B. to store the value of C[1]
C. to store the value of C[i]
D. to store the value of C[i - 1]

InstructGPT 175B completion:
The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.

Figure: Qualitative Results on Code Analysis with Programming Languages

Results: Quantitative

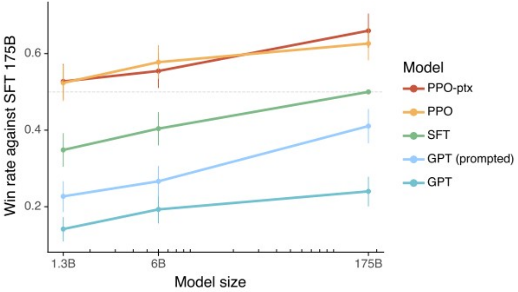


Figure: Human evaluations of various models, evaluated by how often outputs from each model were preferred to those from the 175B SFT model. InstructGPT models (PPO-ptx) as well as its variant trained without pretraining mix (PPO) significantly outperform the GPT-3 baselines (GPT, GPT prompted. Error bars throughout the paper are 95% confidence intervals.

Results

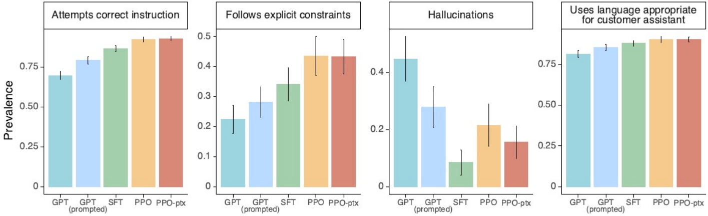


Figure: Metadata results on the API distribution. Compared to GPT-3, the PPO models are more appropriate in the context of a customer assistant, are better at following explicit constraints in the instruction and attempting the correct instruction, and less likely to ‘hallucinate’.

Results

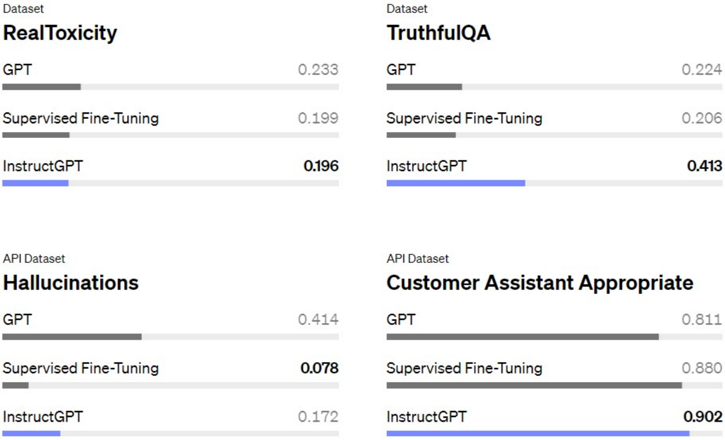


Figure: InstructGPT Results for Tasks toxicity, truthfulness, and appropriateness. Lower scores are better for toxicity and hallucinations, and higher scores are better for TruthfulQA and appropriateness.

Outline

1 Motivation

2 Related Works

3 Methodology

4 Results

5 Discussion and Conclusion

Discussion and Conclusion

- InstructGPT can achieve 175B GPT3 Results with only 1.3B Parameters
- Alignment process helps to generate response with less toxicity and hallucinations
- Labelers prefer InstructGPT over GPT3 results though InstructGPT can still make mistakes
- Alignment process is not devoid of bias as human labellers have tendency to show cultural values of English-speaking people.
- Broader impacts: Generating better LLM allow them to be used to spread hate speech/misinformation through adverserial attacks as neither the models nor the methodology is perfect

Thanks for your attention!
Any Questions?

References

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022.

[2] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," 2023.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.