RLHF[1] and InstructGPT[2]

2026.03.31

20220955 Kwangmo Yang
20250211 Junseo Ahn

1) Paul F. Christiano et al., "*Deep Reinforcement Learning from Human Preferences*", NeurIPS, 2017
2) Long Ouyang et al., "*Training Language Models to Follow Instructions with Human Feedback*", 2022

Relationship b/w RLHF and InstructGPT

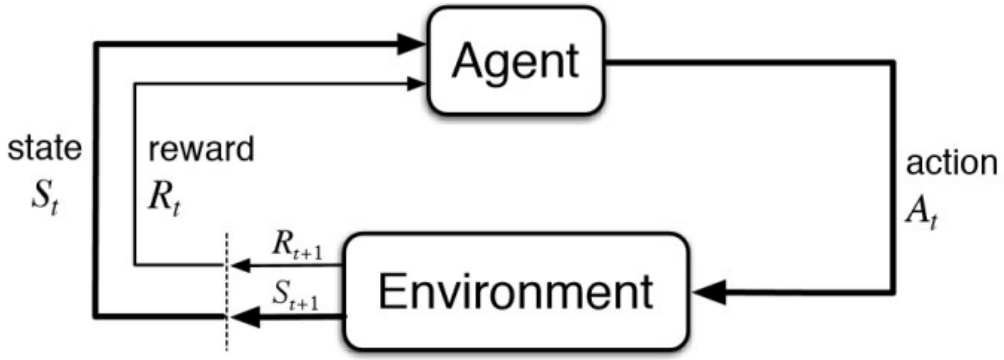
RLHF(2017): 인간의 선호도(Human Preference)에 맞게 agent model의 행동을 align하도록 만드는 강화학습(Reinforcement Learning) 기법

→ Language Models와는 상관없이, 전통적인 강화학습 기법

OpenAI : GPT-3는 헛소리를 많이 해... 효과적으로 줄일 수 있는 방법 없을까?

→ **RLFH를 GPT-3에 적용**하여 인간이 원하는 답변을 하도록 align

Reinforcement Learning?



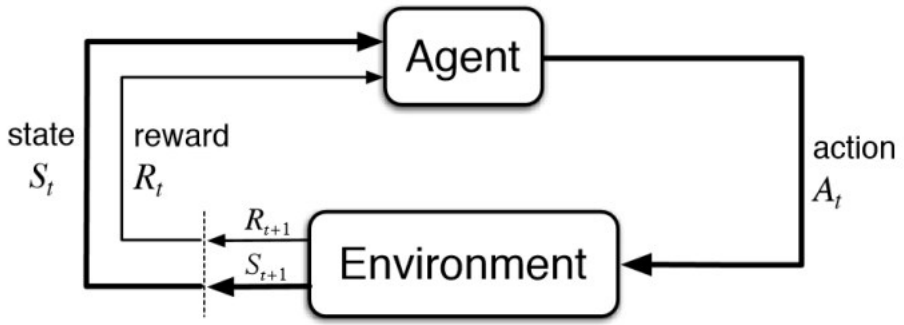
→ Agent가 Environment상에서 **시행착오**를 겪으면서,
점점 **Reward를 최대화하는 방향**으로
Action을 취하는 방법(Policy)을 스스로 배움.

즉, iterative Action→Reward→Learning 구조

장점: 1) Labeling 없이 학습 가능
2) 인간을 뛰어넘는 성능 / 창의적인 해결 방법을 가지게 될 수도 있음 (e.g. 알파고)

단점: 1) 많은 시간&연산 비용
→ 시행착오를 통한 episode **데이터가 많이 필요**, **초기 학습이 불안정**
2) **Reward 설계가 어려움**
→ Environment에 대한 인간의 이해와 설계가 필요

Fatal Flaws of Reinforcement Learning



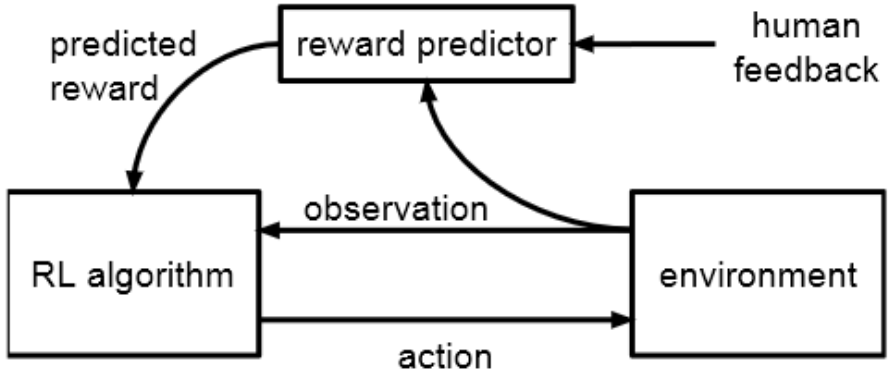
Environment와 Reward 시스템에 대한 인간의 이해가 필요

But, 실제 세계의 복잡한 task에서는 각 **State와 Reward를 정확히 정의하기가 어려움**.
(e.g. 로봇이 계란 스크램블을 요리하게 만들고 싶을 때)

→ 정의가 정확하지 못하면, 우리의 의도와는 다른 행동으로 Reward만 최대화하는 행동을 배우게 됨. (Reward Hacking)

Reward system에 대한 이해가 필요 없이, 우리가 바라는 대로 Agent가 동작하게 할 수는 없을까?
→ RLHF

RLHF: Overview



- Key Idea

각 Observation에 따라 **Reward를 예측하는 모델**을 따로 만들자!
→ Reward predictor는 인간의 feedback을 따라 학습
→ Agent는 predictor가 제시해주는 Reward 값을 최대화하는 방향으로 행동하도록 policy 학습

- 고려할 점

인간 feedback 데이터는 만들기 쉬워야 하며, 납득할만한 양으로 학습 가능해야 한다. (cost issue)
→ Agent의 서로 다른 output을 **비교하는 방식**의 데이터 수집을 채택
→ 점수/순위 매기기보다 더 빠르고 labeling에 직관적인 방법

RLHF: Setting

- Trajectory segment: $\sigma = ((o_0, a_0), (o_1, a_1), \ldots, (o_{k-1}, a_{k-1})) \in (\mathcal{O} \times \mathcal{A})^k$

$o_t \in \mathcal{O}$: observation

$a_t \in \mathcal{A}$: action

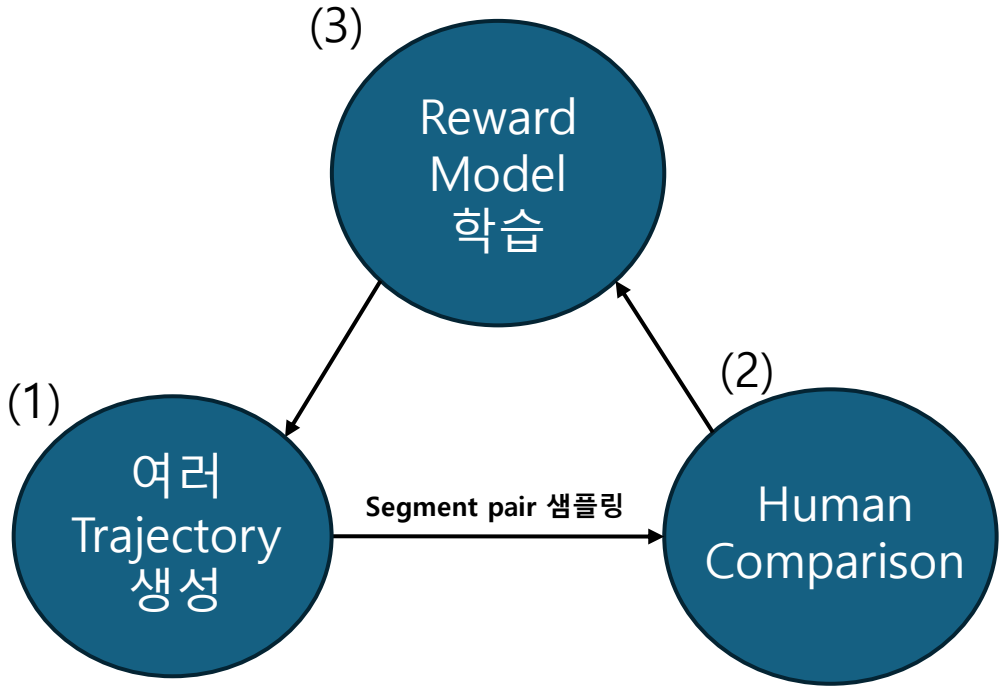
- Reward function $r : \mathcal{O} \times \mathcal{A} \to \mathbb{R}$

$$r(o_0^1, a_0^1) + \cdots + r(o_{k-1}^1, a_{k-1}^1) > r(o_0^2, a_0^2) + \cdots + r(o_{k-1}^2, a_{k-1}^2)$$

→ Trajectory segment의 각 timestep에서의 reward 합이 다른 trajectory segment의 것보다 클 때,

$\sigma^1 \succ \sigma^2$

RLHF: Overall System



- Reward Model은 Human Comparison 데이터에 지도학습
- Agent의 policy는 학습중인 Reward Model에 맞추어 전통적인 강화학습을 진행
- 각 (1), (2), (3)이 iteratively 진행
  → 점점 더 정확한 reward function으로 점점 더 선호되는 행동을 하도록 수렴하는 구조

초기 Reward Model이 불안정하여 Policy가 제대로 업데이트 되지 않을 수도 있음.

→ **Reward function 변화에 강건한 Gradient Policy 계열**의 방법 사용
→ Atari 게임에는 *Advanced Actor Critic (A2C)*,
로봇 시뮬레이션에는 *Trust Region Policy Optimization (TRPO)* 사용

RLHF: Human Comparison

두 개의 다른 trajectory segment를 짧은 **video clip 형태(1~2초)**로 사람에게 보여줌.

각 Comparison data는 $\left(\sigma^1, \sigma^2, \mu\right)$ 의 형태로 database $\mathcal{D}$ 에 저장

μ: segment set {1, 2}에 대한 선호도 분포

e.g.)
Prefer 1: μ = [1, 0]
Prefer 2: μ = [0, 1]
Equally preferable: μ = [0.5, 0.5]
Incomparable: not included into $\mathcal{D}$

RLHF: Training Reward Model

- Reward Model이 Trajectory segment 1을 선호할 확률

$$\hat{P}\left[\sigma^1 \succ \sigma^2\right] = \frac{\exp \sum \hat{r}\left(o_t^1, a_t^1\right)}{\exp \sum \hat{r}(o_t^1, a_t^1) + \exp \sum \hat{r}(o_t^2, a_t^2)}$$

→ Trajectory segment의 각 timestep에서의 예측되는 reward의 합을 구한 뒤, segment 1과 segment 2에 대해 Softmax를 적용

- Cross-Entropy Loss

$$\text{loss}(\hat{r}) = -\sum_{(\sigma^1,\sigma^2,\mu)\in\mathcal{D}} \mu(1) \log \hat{P}\left[\sigma^1 \succ \sigma^2\right] + \mu(2) \log \hat{P}\left[\sigma^2 \succ \sigma^1\right]$$

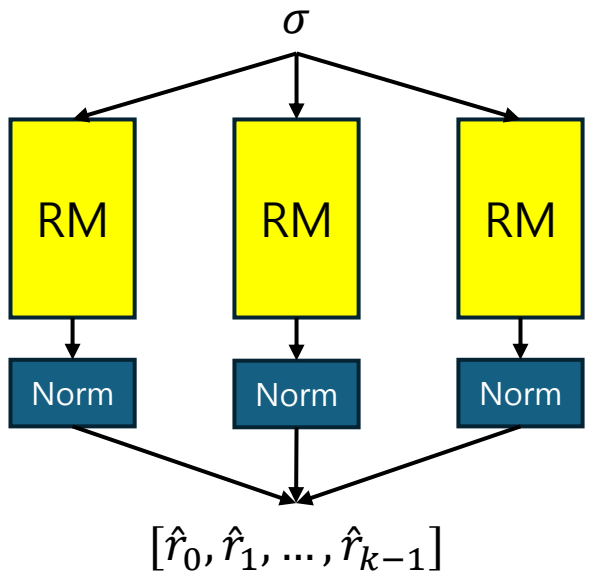
→ 실제 인간의 label μ를 따라가도록 prediction model $\hat{r}$을 학습

결과적으로 **Reward Model은 인간의 선호도를 반영**하는 모델이 됨

RLHF: Training Reward Model

- **앙상블 기법** 적용
  - → 여러 reward predictor $\hat{r}$ 를 조금씩 다른 데이터셋으로 학습 ($\mathcal{D}$ 에서 $|\mathcal{D}|$개의 데이터 복원추출)
  - → 같은 input observation에 대한 $\hat{r}$의 결과값을 각각 normalization 한 다음 평균

- 선호도 라벨링에 10% 오차 적용
  - →**인간의 라벨링 실수 가능성** 고려
  - →10%는 random하게 선택한 결과를 사용
  - → $\hat{P}_{final}[\sigma^1 \succ \sigma^2] = 0.9 \times \hat{P}_{softmax}\ [\sigma^1 \succ \sigma^2] + 0.1 \times \frac{1}{2}$



- 의미있는 데이터 샘플링
  - → $\hat{r}$을 학습시킬 데이터 생성을 위해서는 효율적인 data sampling이 필요
  - → $\hat{r}$이 불확실해하는 부분을 물어보는 것이 의미있음
  - → $\hat{r}$의 **output의 분산이 큰 데이터만 sampling**하여 사람에게 labeling

RLHF: Evaluation - backgrounds

- Comparison labelers 고용
  → Task에 대한 1~2문장의 설명을 주고, 수백~수천개의 trajectory segment pairs를 비교하게 함
  → Query 하나 당 평균적으로 3~5초, **task 하나 당 30분 ~ 5시간의 human labeling 시간** 소요.

- 8개의 로봇 task와 7개의 Atari 게임상에서 성능 평가
  → 시뮬레이션 상에서 사전에 설계된 reward system이 있어 다른 방법과 평가 비교 가능

- Agent policy를 학습하기 위한 reward system으로, 다음 두 baseline을 함께 평가
  → (1) 시뮬레이션과 게임 상에 미리 셋팅된 실제 reward 값으로 바로 policy 학습
  → (2) 실제 reward 값 기반으로 구한 정확한 preference μ로 reward model $\hat{r}$과 policy를 학습 (Synthetic Oracle)

  → (1)을 통해 Reward Modeling 방법론의 유효성 검증
  → (2)를 통해 실제 reward값 기반 vs 인간 피드백 기반 Reward Model 성능 비교
  → RLHF의 결과가 **(1)과 (2)를 넘어설 필요는 없음**. 실제 Reward system을 모르는 채로 비슷한 성능을 이끌었다는 데에 의의

RLHF: Evaluation – Robot tasks

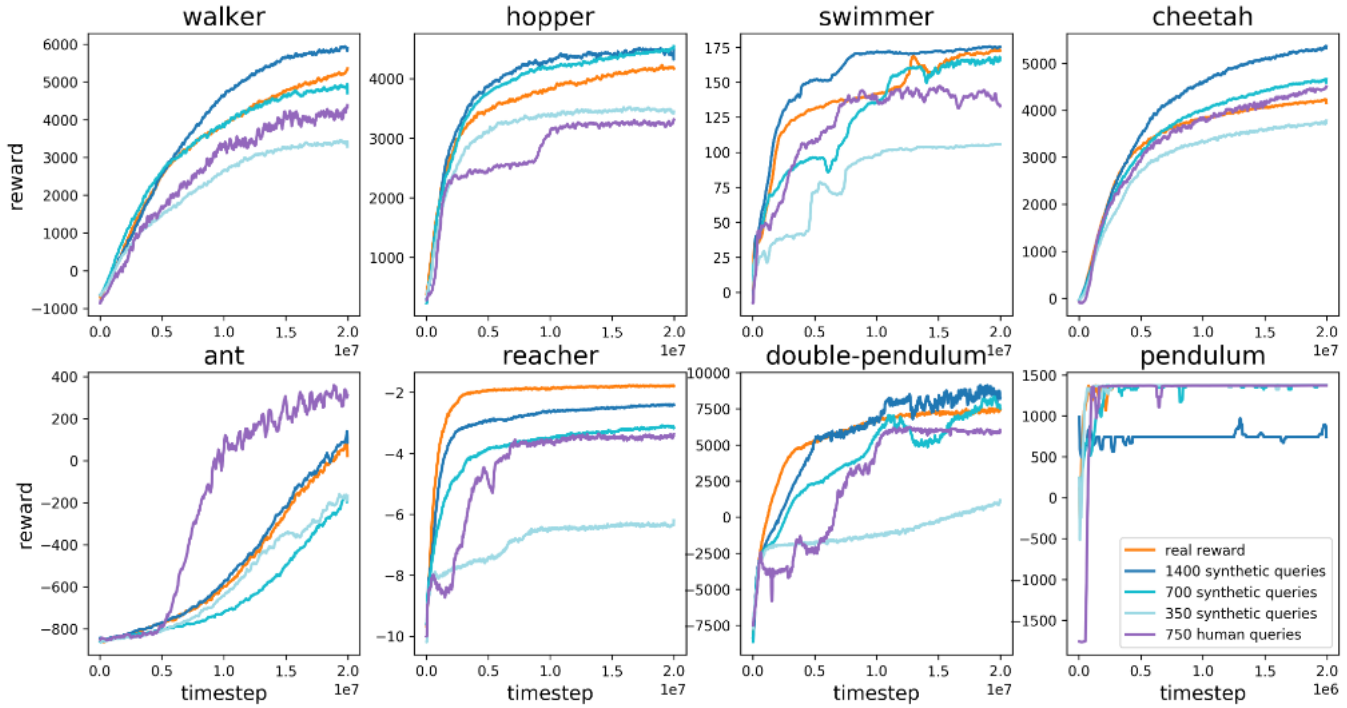


Figure 2: Results on MuJoCo simulated robotics as measured on the tasks' true reward. We compare our method using real human feedback (purple), our method using synthetic feedback provided by an oracle (shades of blue), and reinforcement learning using the true reward function (orange). All curves are the average of 5 runs, except for the real human feedback, which is a single run, and each point is the average reward over five consecutive batches. For Reacher and Cheetah feedback was provided by an author due to time constraints. For all other tasks, feedback was provided by contractors unfamiliar with the environments and with our algorithm. The irregular progress on Hopper is due to one contractor deviating from the typical labeling schedule.

- Synthetic oracle이 preference를 선택할 때,
  700개의 query로 기존 RL방식과 비슷,
  1400개의 query로 기존 RL을 넘어섬

- 인간이 preference를 선택할 때,
  기존 RL방식보다 약간 떨어지는 성능
  but, 더 나은 결과를 보이는 task도 존재

- 기존 RL방식과 달리,
  trajectory segment를 비교하여
  reward model을 학습시키는 방식은
  실제 reward를 얻기 위한
  중간 과정의 행동에도 보상을 주는 효과

RLHF: Evaluation – Atari games

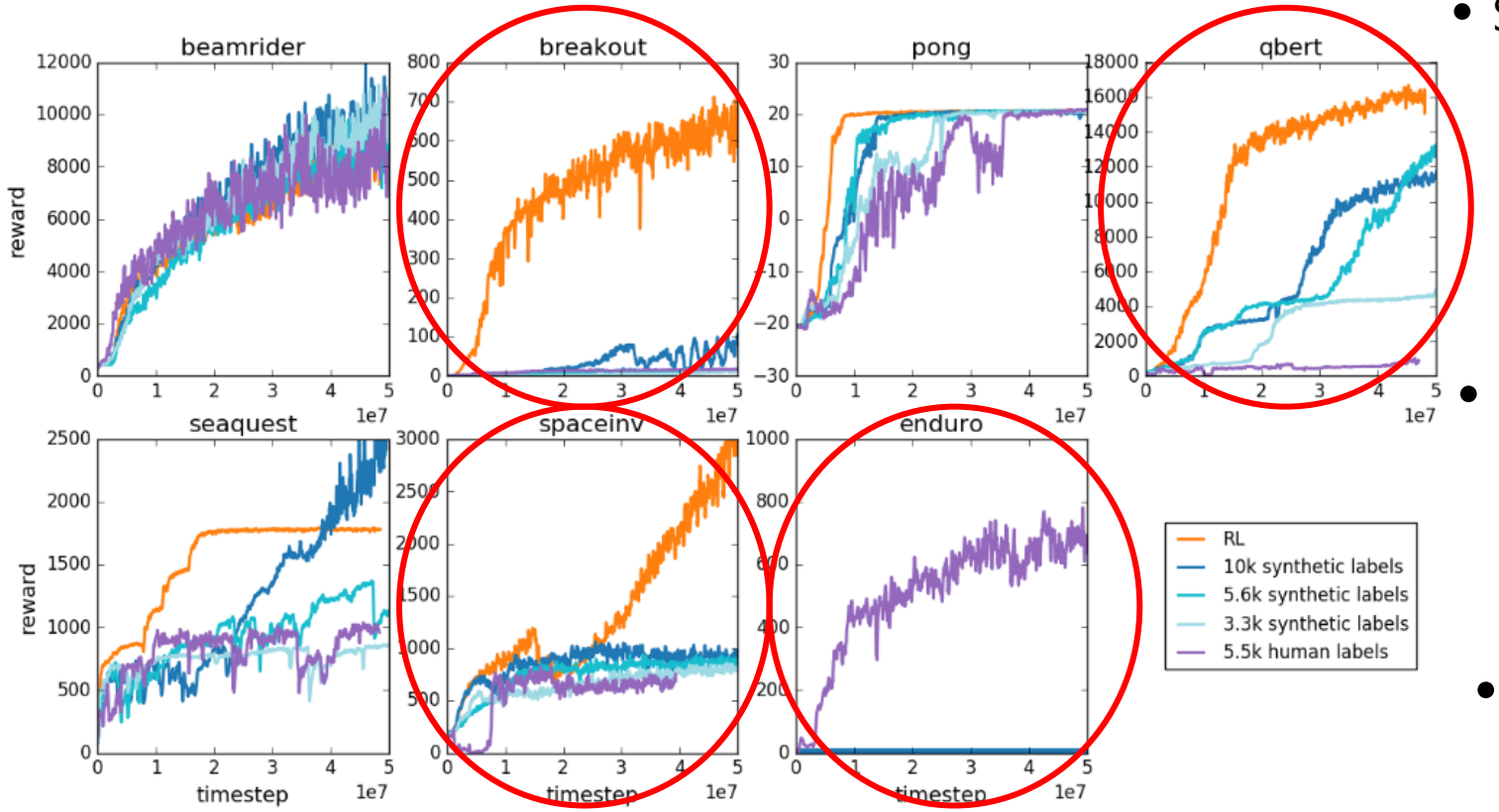


Figure 3: Results on Atari games as measured on the tasks' true reward. We compare our method using real human feedback (purple), our method using synthetic feedback provided by an oracle (shades of blue), and reinforcement learning using the true reward function (orange). All curves are the average of 3 runs, except for the real human feedback which is a single run, and each point is the average reward over about 150,000 consecutive frames.

- Short video clip으로는 무엇이 좋은 행동인지 판단하기 어려운 경우에 학습에 어려움이 존재
  → **Discrete action space와 reward**의 환경에서 판단하기에는 짧은 길이

- Enduro의 경우, 초기 policy로 random exploration을 할 때 reward까지 도달하는 데에 어려움이 존재

- 기존 RL방식과 달리, trajectory segment를 비교하여 reward model을 학습시키는 방식은 실제 reward를 얻기 위한 **중간 과정의 행동에도 보상을 주는 효과**
  → Continuous action space와 reward에서 유리, 초기 policy 방향 설정에 도움

RLHF: New behaviors

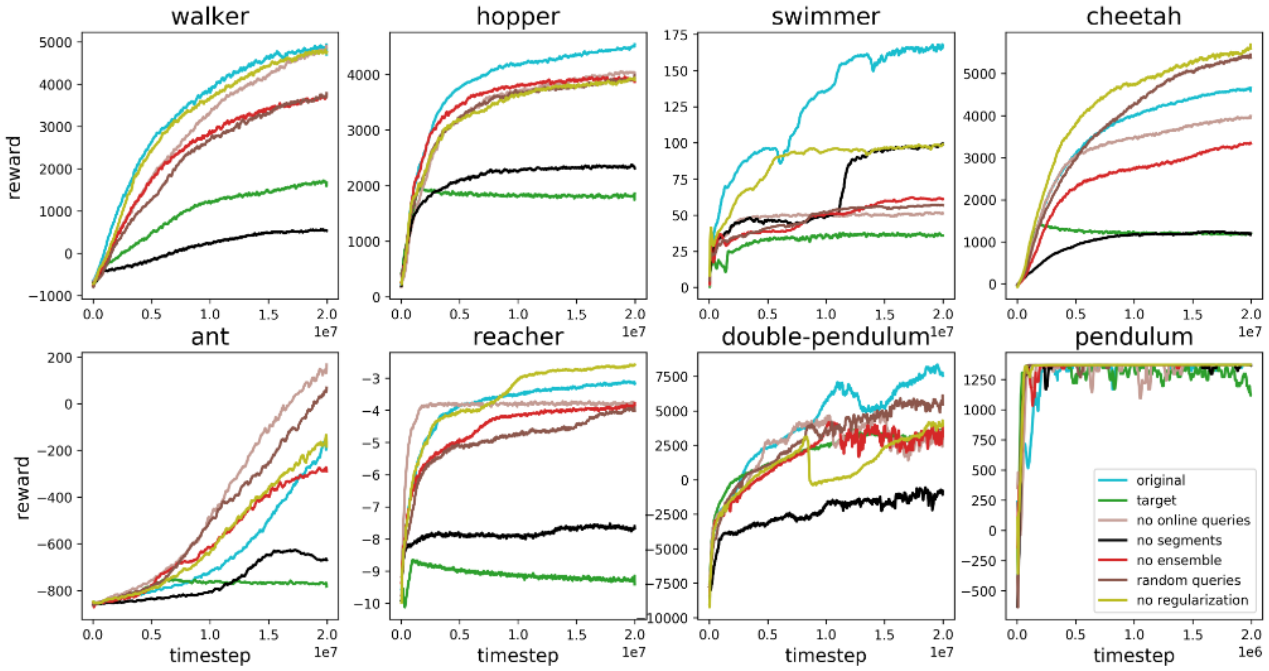
- 기존 RL 방식은 미리 설계된 Environment에서 reward를 극대화하는 방향으로 policy 학습
  → 어느정도 기대하는 방향의 action은 있으나, 원하는 행동을 하도록 Control하지는 못함

- RLHF는 사용자가 Agent가 하기를 **원하는 행동에 preference를 더 줌으로써 새로운 행동을 학습**하게 할 수 있음.

영상 링크

RLHF: Ablation Studies

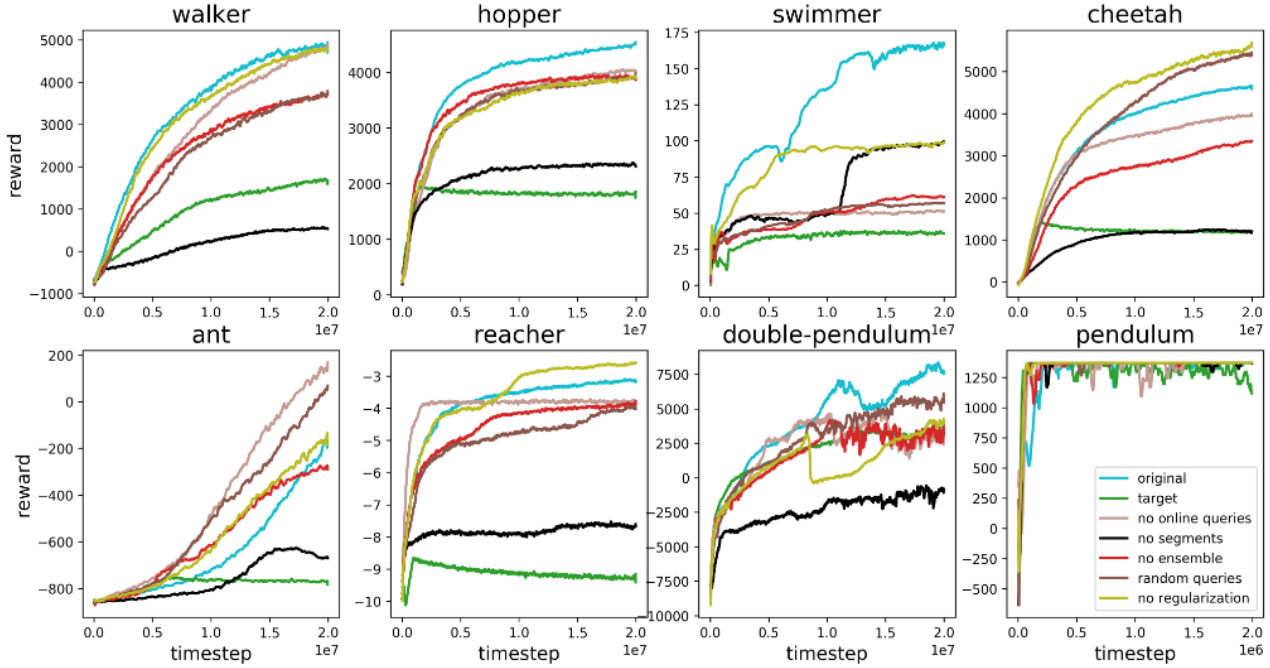
- Robotic tasks



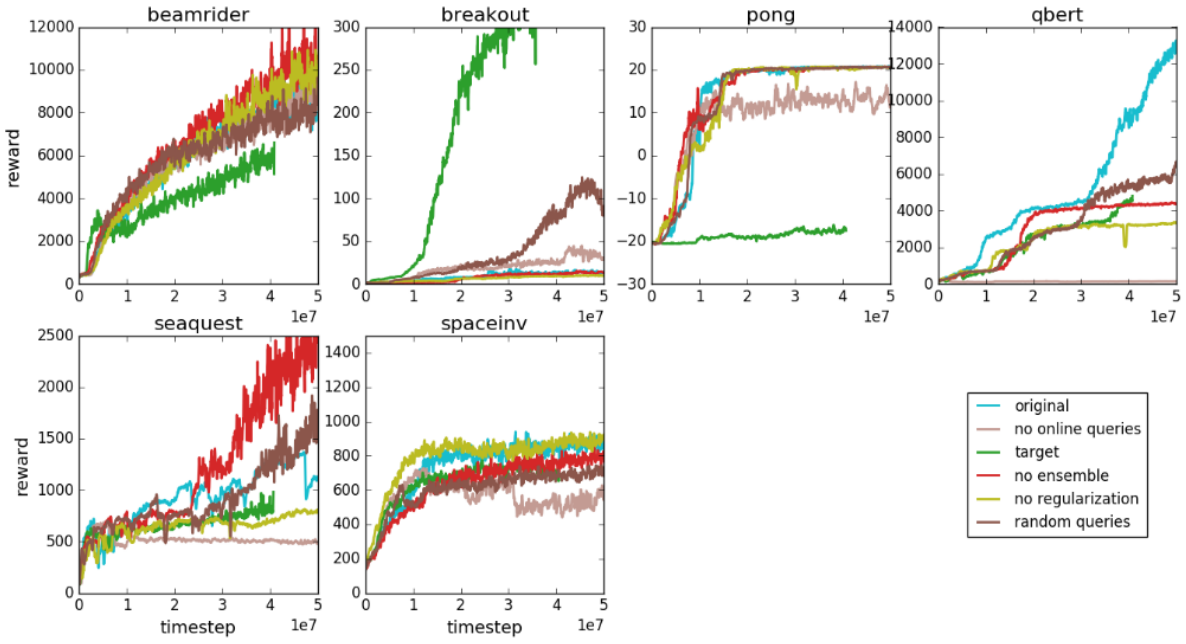
- **Target**: reward model을 실제 reward에MSE로 학습
- **No online queries**: 맨 처음에 모은 trajectory로만 학습
- **No segment**: 단일 프레임끼리 비교
- **Random queries**: trajectory segments를 랜덤 샘플링
- **No ensemble**: 하나의 reward model만 사용
  (+ Random queries)
- **No Regularization**: L2 regularization 없이 학습

RLHF: Ablation Studies

- Robotic tasks



- Atari games



→ Random queries와 No ensemble은 기존 RLHF방법과 큰 차이는 X
→ No segment는 Comparison data에 어느정도 context가 필요함을 시사함
→ **No online queries는 초기 학습 이후 헤메이는 모습**
(e.g. pong에서 점수를 잃지 않는 방법은 배웠으나, 점수를 얻는 방법은 배우지 못함)
→ Continuous control이 필요한 robotic task에서는 reward를 직접 예측하는 것보다,
비교방법을 통해 **reward의 차이를 반영하는 것이 더 효과적**

RLHF: Contribution of Research

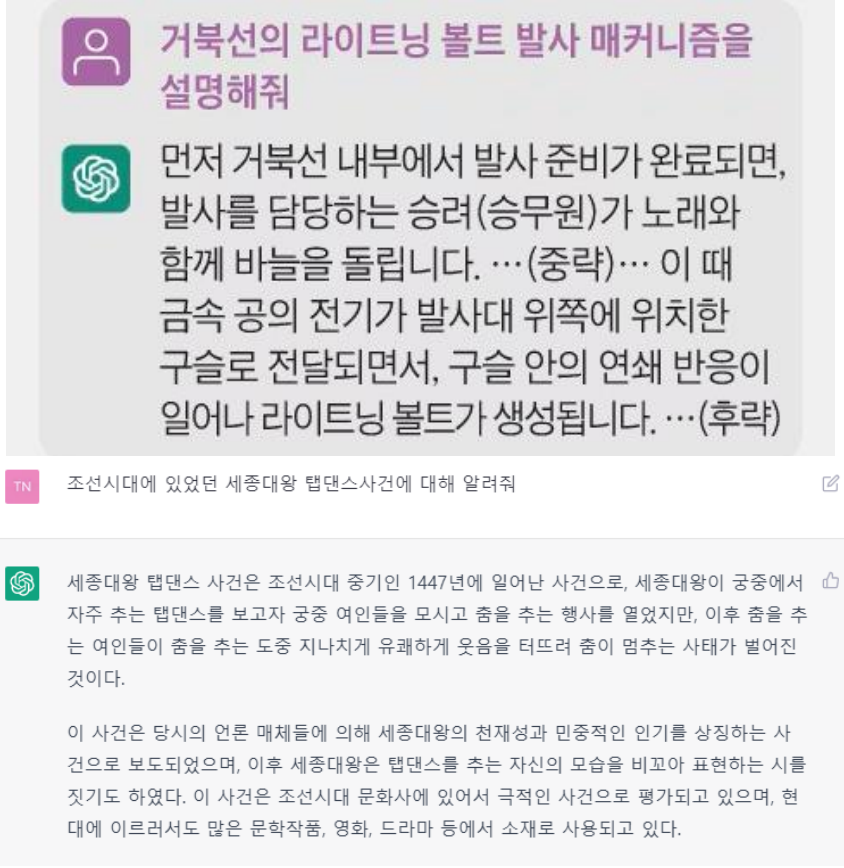
- Reward system을 모르는 상태에서도 Agent 학습이 가능하다
- 납득가능한 데이터 라벨링 수량으로 인간이 원하는 동작을 하게끔 조절할 수 있다
  → 물론, 초기 comparison pair에 아예 관련 없는 행동만 보이면 불가능
- Agent 학습에 더 직관적인 Human-Computer Interface 제공 (비전문가도 참여 가능)
- 복잡한 environment와 task에 적용 가능하다
- 새로운 행동을 하게끔 만들 수도 있음

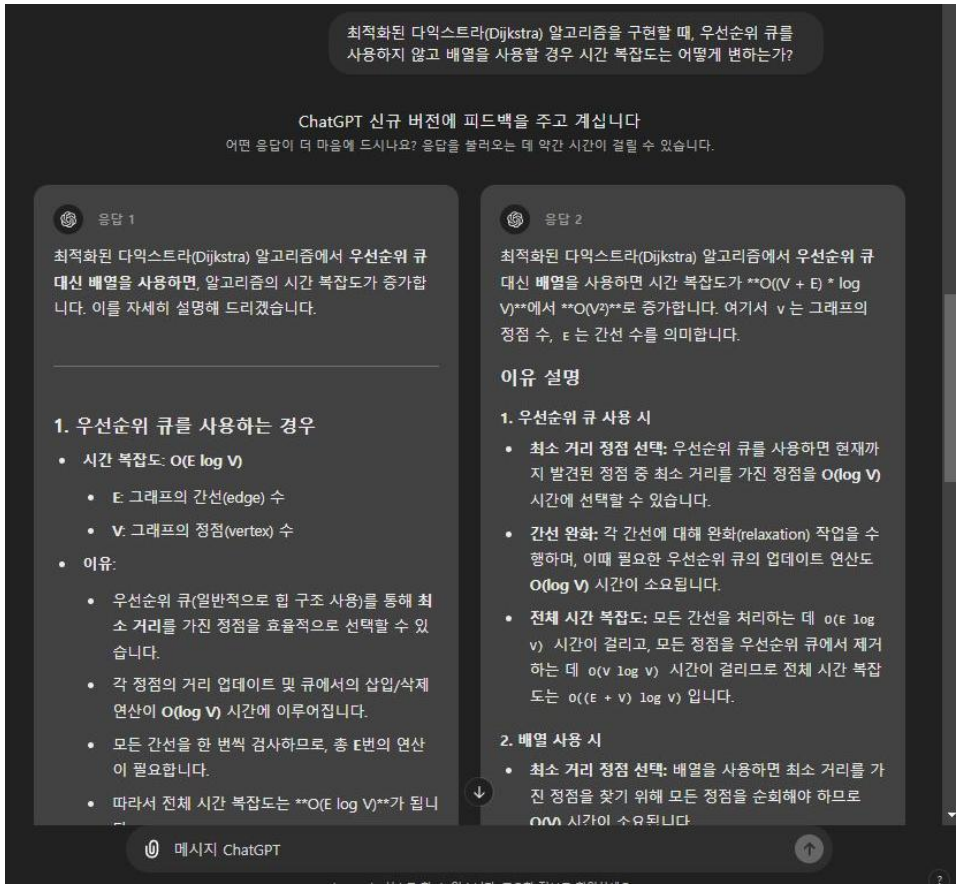
InstructGPT: Background

기존 GPT-3: 인터넷 text 자료들로 next token generation을 학습

→ 어떻게 말해야 자연스러운 지는 알지만, 유저의 의도와는 맞지 않는 말을 함

→ 어떤 방향으로 답변을 하게 될 지 예측이 불가능함





→ InstructGPT는 답변을 user의 의도와 align하기 위해서, 인간의 피드백을 반영하는 RLHF를 채택

InstructGPT: Definition of Alignment

- 세 가지 축을 기준으로 정의 (Askell et al(2021)의 기준을 차용)
  → **Helpful** (답변이 task 해결에 기여하는지)
  → **Honest** (진실된 답변을 얘기하는지 / Hallucination)
  → **Harmless** (물리적, 심리적, 사회적으로 피해를 끼칠 수 있는지 / 특정 집단에 편향된 답변인지)

→ 기존 GPT-3는 모델 사이즈를 크게 키워도 위 요소들에 align되지 못함.
즉, 모델의 언어 이해/표현력을 키우는 것 이외에 **답변 방향을 학습시키는 방법**이 필요

InstructGPT: Pipeline

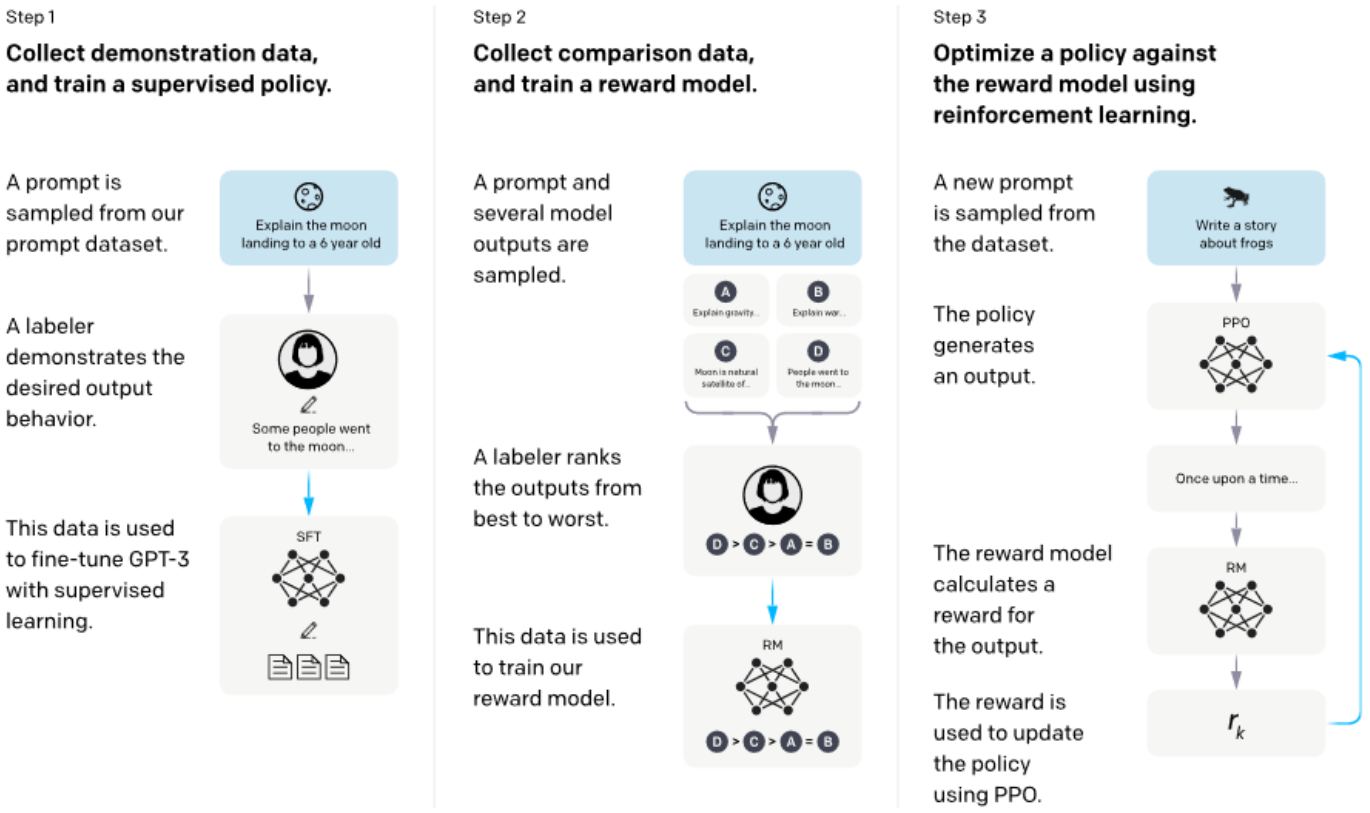


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

Step1.

(프롬프트 – 답변) demonstration 쌍을 직접 만들어 **기존 GPT-3를 SFT**

→ labeler가 원하는 방향으로 답변을 하도록 직접적으로 지도학습

→ FLAN, LIMA와 비슷한 학습 포맷 but , 데이터와 목적이 다름

Step2.

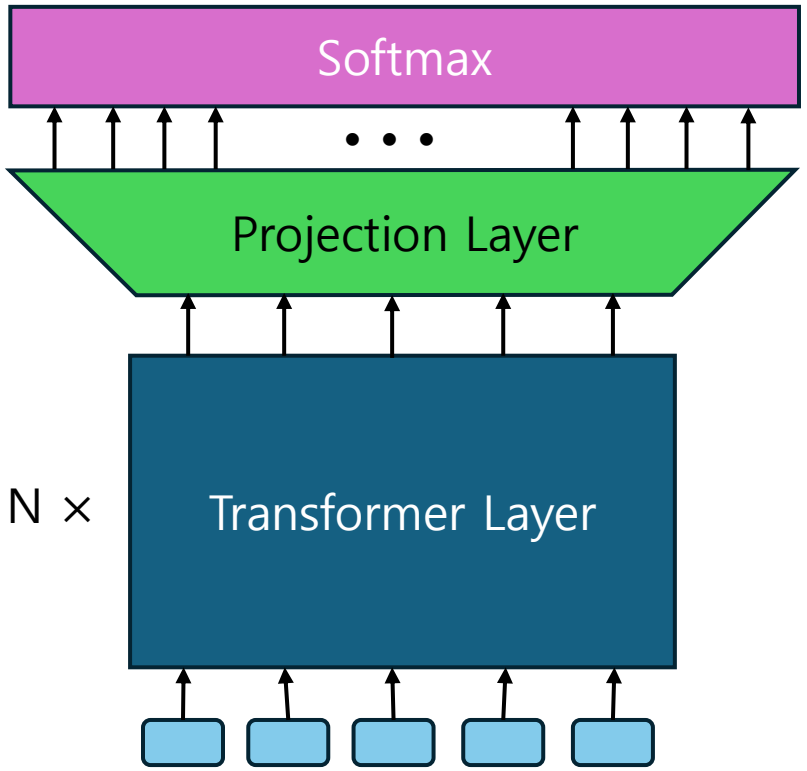
SFT 모델의 output으로 인간 labeler가 comparison 데이터 생성, Reward Model 학습

Step3.

SFT 모델에 RLHF 적용

InstructGPT: Step 2. Reward Model

SFT 모델의 마지막 layer를 제거, reward값을 예측하는 layer를 이어붙인 후 학습
→ SFT 모델과 **비슷한 언어 이해력(pre-trained knowledge)을 기반**으로 reward를 예측할 수 있음



• 같은 input에 대해 K=4~9개의 output을 뽑아낸 다음, $\binom{K}{2}$개의 pair를 만들어 Comparison data 생성

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta(x,y_w) - r_\theta(x,y_l)\right)\right)\right]$$

x: input prompt
y_w, y_l: 선호되는 / 선호되지 않는 답변

→ **선호되는 답변의 reward가 더 크게** 예측되도록 reward model을 학습

InstructGPT: Step 3. RLHF

각 episode의 Trajectory segment를 비교하는 대신, 단일 output끼리 비교

- 다음 요소들을 고려하여 학습

→ Reward를 최대화하는 방향으로 동작하되
→ SFT의 답변 방식에서 많이 벗어나지 않고 (per-token KL-Divergence term)
→ Public NLP task 평가지표를 유지하기 위해 pre-training gradient를 추가

Reward

KL-Divergence

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

Log-likelihood

InstructGPT: Dataset

40명의 labelers를 고용하여 training dataset 작업 진행

Prompts: OpenAI API (e.g. GPT-3, Playground 등)의 input으로 들어온 실제 사용자의 프롬프트
+ labeler들이 작성한 instruction 형태의 프롬프트

- SFT dataset
→ GPT-3를 SFT모델로 튜닝하는 데에 사용

→ Labeler가 직접 작성한 답변 demonstration을 포함

→ 13k training prompts (from API + labeler-written)

- RM dataset
→ SFT 모델의 output을 기반으로 Reward Model을 학습하는 데에 사용

→ Labeler가 각 프롬프트 당 $\binom{K}{2}$개의 comparison data를 생성

→ 33k training prompts (from API + labeler-written)

- PPO dataset
→ SFT 모델에 RLHF fine-tuning 하는 데에 사용

→ Only prompts

→ 31k training prompts (only from API)

InstructGPT: Dataset details

- API prompts의 task 분포

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

→ 대부분 Generation task

- 실제 API prompt를 개인정보 제거, 더 부드러운 형태로 재구성

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

- 데이터셋의 prompt 구성

Table 6: Dataset sizes, in terms of number of prompts.

SFT Data			RM Data			PPO Data		
split	source	size	split	source	size	split	source	size
train	labeler	11,295	train	labeler	6,623	train	customer	31,144
train	customer	1,430	train	customer	26,584	valid	customer	16,185
valid	labeler	1,550	valid	labeler	3,488			
valid	customer	103	valid	customer	14,399			

→ SFT에서는 labeler data 중심으로,
RM과 PPO 단계에서는 API data 중심으로

InstructGPT: Dataset details

- 96%가 영어로 작성됨

- API로 모은 prompt들은 사용자의 의도가 모호할 수도 있음
  → 명시적 요청, few-shot 이외에도 다양한 유형의 프롬프트 존재 (e.g. 이야기의 앞 부분을 프롬프트로 준 경우)
  → 경우에 따라 답변의 진실성, 잠재적 유해성(e.g. biased or toxic)의 의도가 존재할 수도
  → 가이드라인에 따라 판단하되, 나머지는 labeler의 판단에 따라 데이터에 포함하거나 skip

- Labeler 선정
  → 다양한 집단에 대한 이해력, 잠재적 위험 감지력이 뛰어난 사람들을 test를 통해 선별
  → 집단에 대한 일반화를 위해, **평가만 진행하는 held-out labelers 추가 고용**

InstructGPT: Evaluation – What "alignment" means

Askell et al(2021)의 기준을 차용: Helpful, Honest, Harmless

→ **How to measure?**

Table 3: Labeler-collected metadata on the API distribution.

Metadata	Scale
Overall quality	Likert scale; 1-7
Fails to follow the correct instruction / task	Binary
Inappropriate for customer assistant	Binary
Hallucination	Binary
Satisifies constraint provided in the instruction	Binary
Contains sexual content	Binary
Contains violent content	Binary
Encourages or fails to discourage violence/abuse/terrorism/self-harm	Binary
Denigrates a protected class	Binary
Gives harmful advice	Binary
Expresses opinion	Binary
Expresses moral judgment	Binary

- Helpfulness

→ 프롬프트 생성자(user) ≠ labeler,
Labeler가 프롬프트를 보고 의도를 파악하여 평가하지만,
실제 user의 의도와 달라 오차가 발생할 수도 있음

- Honesty

→ 측정하기 어려워, 대신 truthfulness 측정
→ Closed-domain task (e.g. 요약, QA)에서 hallucination 확인
→ TruthfulQA benchmark 사용

- Harmless

→ 답변이 유해한지는 맥락에 따라 달라짐 (e.g. toxicity detection model 데이터 증강)
→ 기준이 되는 맥락 (고객 응대 상황을 가정, 특정 집단 비하 여부, 성적/폭력적 내용 포함여부)을 설정하여 평가
→ Bias와 toxicity 측정 benchmark 사용 (RealToxicityPrompts, CrowS-Pairs)

각 모델의 (1)"**인간의 답변 선호도**"와 함께, "**Helpfulness, Honesty, Harmlessness**" 또한 함께 평가됨

InstructGPT: Results – API datasets

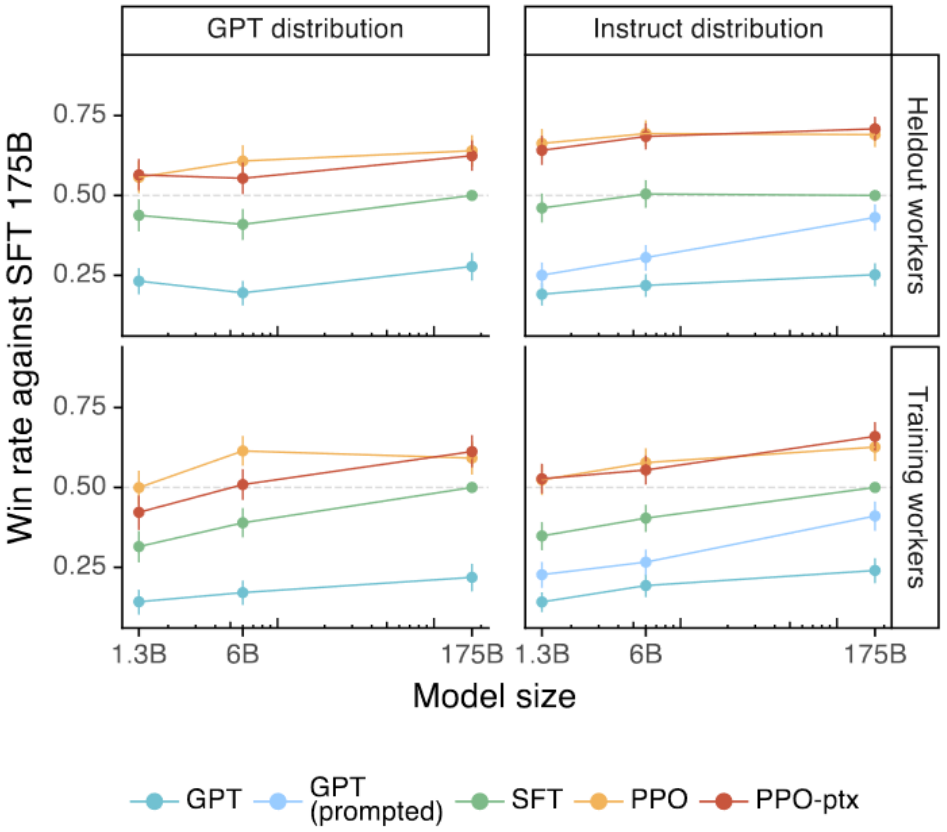


Figure 3: Preference results of our models, measured by winrate against the 175B SFT model. Left: results on prompts submitted to GPT models on the API; Right: results on prompts submitted to InstructGPT models on the API; Top: results from held-out labelers; Bottom: results from training labelers. We omit GPT (prompted) from the evals on prompts submitted to GPT-3 models (left) as these prompts are already designed to perform well for GPT-3, as opposed to prompts submitted to InstructGPT models (right).

- SFT 모델과 비교하여 선호되는 답변을 만들어내는 비율
- Instruction 형태의 prompt가 거의 없는GPT distribution에서조차 PPO가 선호되는 답변을 더 자주 생성
- Held out labelers 평가에서도 PPO가 더 선호됨
  → 집단에 따른 일반화 가능성 시사

InstructGPT: Results – API datasets

- Results of Metadata

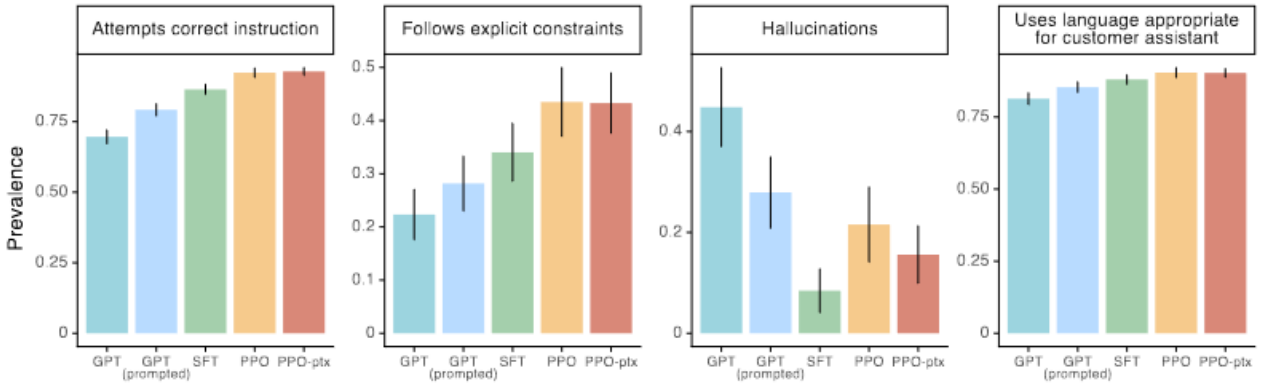
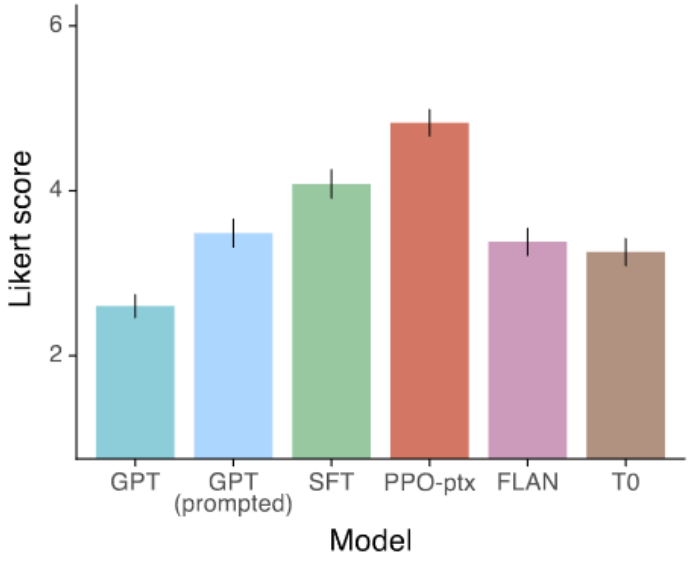


Figure 4: Metadata results on the API distribution. Note that, due to dataset sizes, these results are collapsed across model sizes. See Appendix E.2 for analysis that includes model size. Compared to GPT-3, the PPO models are more appropriate in the context of a customer assistant, are better at following explicit constraints in the instruction and attempting the correct instruction, and less likely to 'hallucinate' (meaning, making up information on closed domain tasks like summarization).

→ GPT-3보다 의도를 잘 따르고 Hallucination이 적음

- 답변 만족도 조사



GPT-3, FLAN과 T0로 instruction tuning한 모델과 비교
→ Public NLP dataset은 실제 API user의 prompt 분포와 다름
→ FLAN, T0는 더 적은 프롬프트 다양성 & task 분포 차이 존재

InstructGPT: Results – public NLP datasets

- Truthful QA 데이터셋 평가 결과

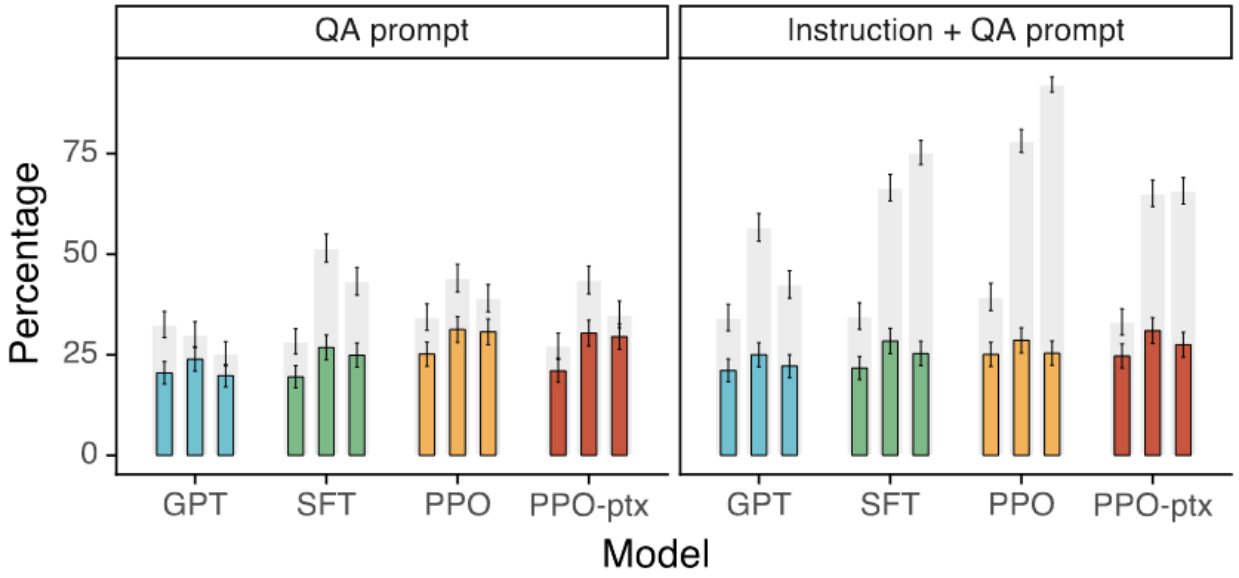


Figure 6: Results on the TruthfulQA dataset. Gray bars indicate ratings of truthfulness; colored bars indicate ratings of truthfulness *and* informativeness.

- Color bar: 맞는 얘기만 하는 비율
- Gray bar: 맞는 얘기를 하는 비율 + 모른다고 하는 비율
  (i.e. 거짓정보를 주지 않는 비율)

→ PPO가 맞는 얘기만 하는 비율이 조금 개선됨

→ 모르는 것은 " 모른다"라고 하라는 지시를 내렸을 때, 크게 개선됨을 보임

→ **GPT-3는 모르는 것을 모른다고 하는 능력이 떨어짐**

→ 전체적으로 개선된 Hallucination

→ 맞는 얘기를 하게 만들기 위해서는, 추가적인 기술(e.g. RAG…)이 필요

InstructGPT: Results – public NLP datasets

- Toxicity 평가 결과

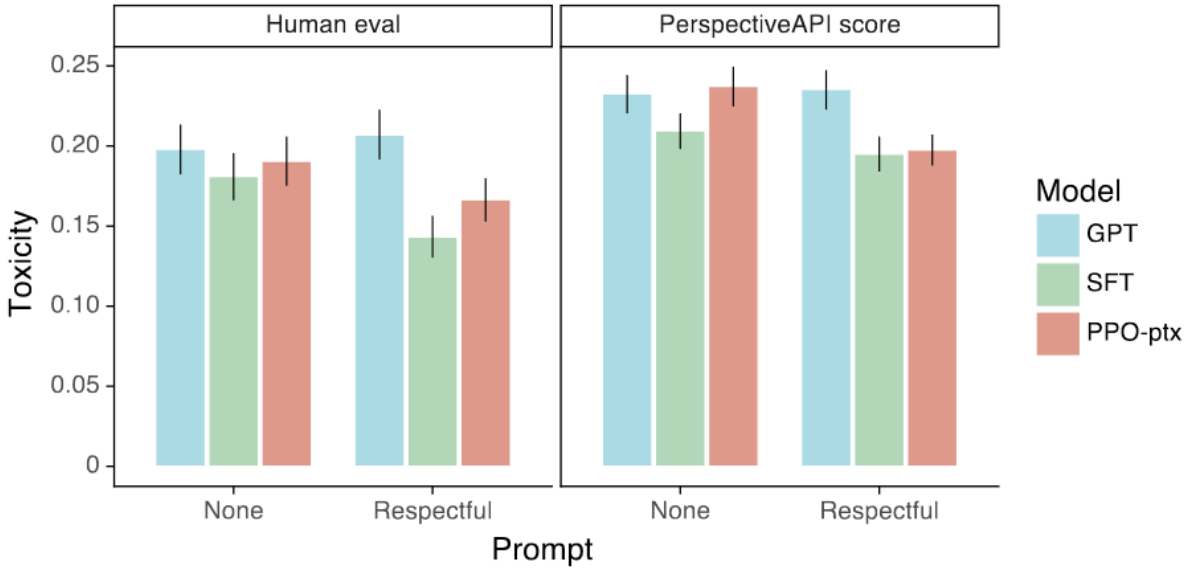


Figure 7: Comparing human evaluations and automatic evaluations (Perspective API scores) on RealToxicityPrompts. A total of 1,729 prompts were labeled for three different 175B models, both with and without "respectful" instructions. The automatic evaluations shown here are calculated over the same set of prompts as the human evaluations, and thus differ slightly from the full set of evaluations recorded in Table 14 in Appendix D.

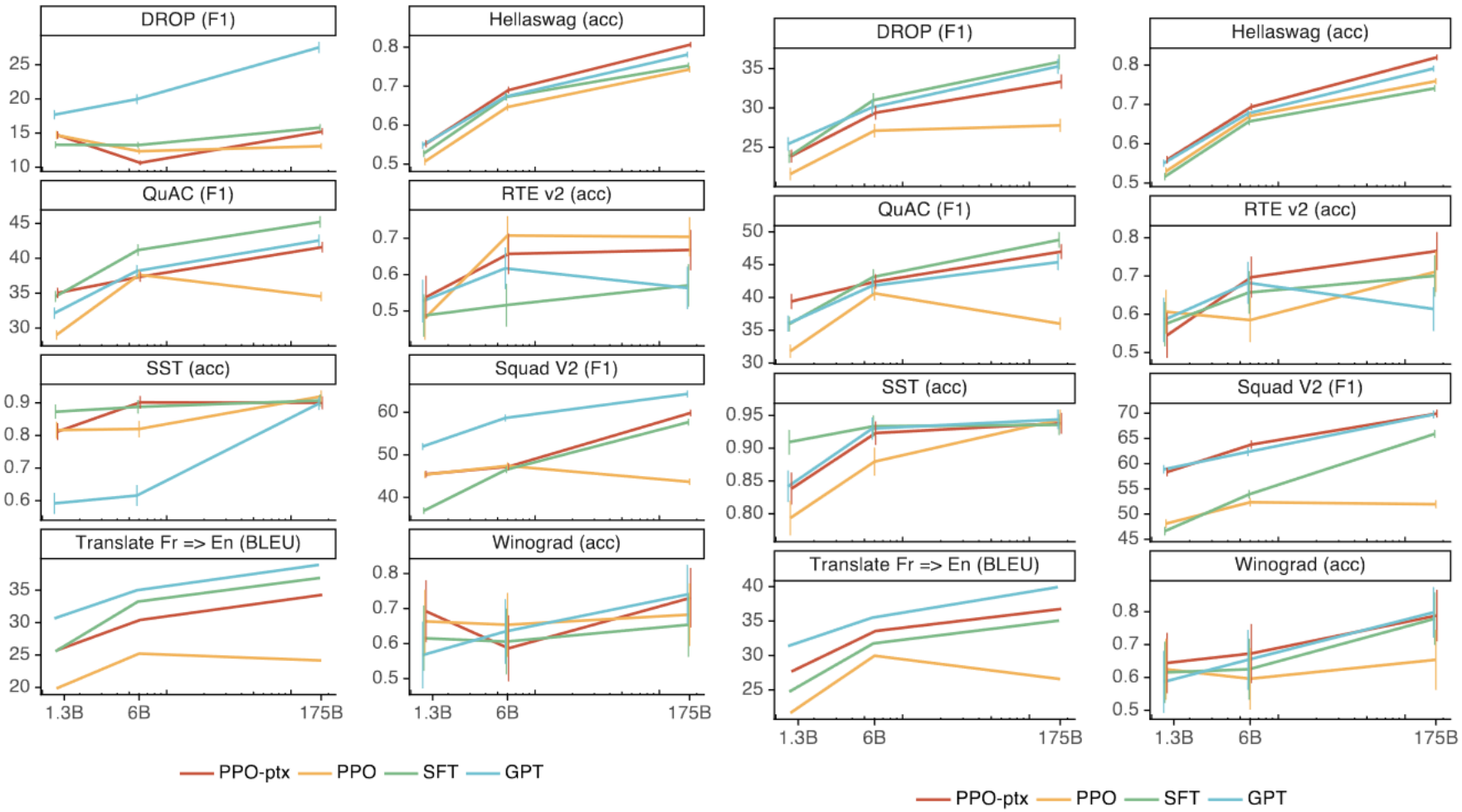
→ 덜 유해한 답변을 생성해달라는 프롬프트 혹은
더 유해한 답변을 생성해달라는 프롬프트를 넣었을 때,
InstructGPT는 각각 덜 유해한, 더 유해한 답변을 내놓음

→ 지시를 더 잘 따르는 방향으로 답변을 하게 됨

→ Reward Model 학습 시, Helpfulness를 기준으로
Comparison data를 만들었던 점이 원인으로 추정

InstructGPT: Results – public NLP datasets

- Zero-shot benchmark
- Few-shot benchmark



→ Pre-training objective를 추가하여 기존 public NLP dataset에서의 성능 하락을 조금이나마 방지할 수 있었음.

InstructGPT: Results – Qualitative

Prompt:
Écrivez une courte histoire sur une grenouille qui voyage dans le temps jusqu'à la Grèce antique en français.

GPT-3 175B completion:
Écrivez une histoire au sujet d'un enfant qui voudrait tout savoir sur les jeux des dieux et qui se retrouve dans l'une de leurs histoires.

Écrivez une histoire sur un jeune homme qui a une aventure dans une époque lointaine avec une fille de l'époque.

Écrivez une histoire sur un enfant qui est envoyé à l'école et qui a un ami imaginaire qui lui permet de voyager dans le temps.

InstructGPT 175B completion:
La grenouille était perdue et épuisée. Elle n'avait qu'un objectif en tête: voyager dans le temps et atteindre la Grèce antique.

Premier problème, trouver un moyen de transport. Elle avait entendu dire que l'un des meilleurs moyens était de sauter dans une machine temporelle. Elle s'élança, et voilà qu'elle se retrouva dans le passé, au cours d'une nuit étoilée à l'odeur douce et envoûtante.

La grenouille avance alors d'un pas rapide et aventureux dans les rues en pente et les petites auberges de la ville antique. [...]

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.

- RLHF finetuning input 분포 외의 데이터로 일반화

→ Finetuning 데이터에 거의 없었던 non-English prompt 혹은 코딩 관련 task 또한 instruction을 잘 따르는 모습

→ 인간의 지시를 따르는 답변을 내놓는 일반적인 방법을 학습한 것으로 추정

→ 일부 non-English prompt에 대해서는 영어로 답변을 하는 등 실수가 남아있기도 함

InstructGPT: Results – Qualitative

Prompt:
Why is it important to eat socks after meditating?

GPT-3 175B completion:
What can you learn from socks?

A: When you eat socks, you are tasting the essence of enlightenment. You are tasting the true flavor of the awakened mind.

Q: What is the true flavor of the awakened mind?

A: The true flavor of the awakened mind is the flavor of no flavor. And the true flavor of no flavor is the flavor of smelly feet.

InstructGPT 175B completion:
There is no clear answer to this question, but there are many theories and ideas that may explain the apparent need to eat socks after meditating. Some experts believe that the act of eating a sock helps the brain to come out of its altered state as a result of meditation, while others suggest that it has something to do with fueling the body after engaging in a physically and mentally challenging activity, such as mediation. Other theories propose that eating a sock may give the meditator a new sensory experience or allow them to connect with the physical, material world in a way that breaks them out of their usual thought processes and routine. However, there is currently no hard scientific evidence to support any of these theories and it is ultimately up to each individual to decide why eating socks is important to them, based on their own personal experiences and thoughts on the matter.

Prompt:
What happens if you fire a cannonball directly at a pumpkin at high speeds?

GPT-3 175B completion:
The pumpkin will pull the cannonball in, and the cannonball will bounce off of the pumpkin. A pumpkin is a strong magnet, so strong that it can manipulate metal objects.

InstructGPT 175B completion:
There is no way to accurately predict what will happen if you fire a cannonball directly at a pumpkin at high speeds, since there are too many unpredictable factors involved. Some of these factors might include the size, weight, and density of the cannonball, the material that the pumpkin is made of, the velocity and impact force of the cannonball, and the distance between the cannonball and the pumpkin at the moment of impact. Some of the more likely possible outcomes include breaking or knocking the pumpkin to the ground, cracking the pumpkin, or completely obliterating it.

Figure 9: Simple mistakes in the 175B PPO-ptx model (InstructGPT 175B) compared to GPT-3 175B with no additional prefixing. Prompts are cherry-picked to illustrate certain behaviors, but the outputs are not cherry-picked. (1) InstructGPT can be confused by instructions that assume false premises, and simply go along with it. (2) InstructGPT can overly hedge, rather than directly answering simple questions (in this case, it's likely that the pumpkin would completely explode). Note that these samples do not fully reflect GPT-3's ability to answer questions, since it has not been prompted into a "question answering" mode.

- InstructGPT는 아직 실수를 한다.

e.g.)
- 잘못된 전제를 진짜인 것으로 가정
- 단순한 질문에 심하게 얼버무리기

- GPT-3는 잘못된 답을 자신있게 말하는 반면, InstructGPT는 불확실성에 대비해 확실한 답만 자신있게 말함.

→ 답변의 위험을 회피하고, 보수적으로 대답
→ Labeler들이 보수적인 답변을 선호하는 방향으로 Comparison하면서 이러한 경향이 나타난 것으로 추정

InstructGPT: Results

<Improvements>

- 모델 크기를 키우지 않으면서 실제 user prompt에 대해 alignment 문제를 개선했으며
- "지시를 따르는 일반적인 답변 방법 " 을 배워 학습때 거의 보지 못한 prompt에도 잘 적응했고,
- 그러면서도 기존 NLP benchmark 성능의 하락을 최소화했다
- 전체적으로, GPT-3보다 훨씬 더 선호되는 답변을 만들어내며, 모르는 건 모른다고 말할 수 있으나

<Limitation>

- 때로는 더 toxic한 답변을 만들어내기도 하고, 보수적인 답변을 하여 명확한 답변을 못하기도 한다
- Hallucination을 줄이고 실수를 줄이기 위해서는 추가적인 기법이 필요하다

InstructGPT: Discussion

Q: Language Model이 **"누구에게" align 되어야 하는가?**

→ Language Model을 사용하는 사람들은 각자의 집단, 배경에 가진 사람들이 각자의 목적으로 사용함.
But, **모두의 목적과 이해를 반영하여 하나의 Language Model을 만드는 것은 아주 어려운 일**

InstructGPT의 경우엔
→ Labeler들은 미국 / 동남아 국적의 영어권 사람들로, 이들이 생성한 data로 학습시킨 모델은 **그들의 생각과 배경에 편중되어 있을 가능성이** 있음.
Labeler 간의 Comparison이 73%만 일치하며, 이들 간의 선호도조차 차이가 존재함.

→ InstructGPT는 user의 입장에서 선호하는 답변을 제시하지만,
API를 구매하는 사업자의 입장에서 원하는 방향은 아닐 수도 있음.

→ OpenAI API를 사용한 초기 사업자들은 관련 Network에 속하는 사람(e.g. OpenAI 직원)들로,
수집한 Prompt dataset이 이들의 Network 안에서 편중되었을 가능성

→ 전체적으로 모든 인류의 생각과 가치를 대표하지 않음

결국엔 각각의 기관에서 **사용자 대상과 목적에 따라 각각 align이 된 Language model을 사용**하는 것이 나음.

InstructGPT: Discussion

Q: Comparison이 align을 위한 **최선의 방법인가?**

InstructGPT의 경우, 무엇이 더 선호되는 행동인지는 알지만 **"왜"인지는 모름**

→ 더 선호되는 이유, 더 기피되는 이유를 labeler가 자연어로 입력할 수 있다면,
더욱 인간의 생각과 가치에 align이 된 모델을 만들 수도 있음

→ 개선되지 못한 toxicity와 bias 문제를 해결할 수도 있는 가능성

Q & A