Reinforcement Learning from Human Feedback (RLHF)

基于人类反馈的强化学习

[M]s 昇思 MindSpore

Contents

Recap: Instruction Tuning

Instruction Finetuning

Pros:

- Simple and straightforward, generalize to unseen tasks

Cons:

- **Expensive** to collect gound-truth data for tasks
- No right answer to tasks **open-ended creative generation**
- **Equal penalization** to all token-level mistakes (even though some are worse than others)

still cannot maximize human preferences

[M]^S 昇思 MindSpore

Recap: Instruction Tuning

- **Limitation in tasks like open-ended creative generation:**
  - *Write me a story about a dog and her pet grasshopper.*

- **Difficult to create loss function for abstract objectives:**
  - *what is funny?*
  - *what is ethical?*
  - *what is safe?*

Non-RLHF vs RLHF

- **Human Feedback** in conversations

- acknowledges the repetition and apologizes for it.
- provides more detailed information about the recommended movies

Conversation with a Machine without RLHF:

User: Hello, can you recommend a good movie to watch tonight?

Machine: Sure! What genre are you in the mood for?

User: I'm in the mood for a comedy.

Machine: How about "The Hangover"? It's a hilarious comedy that follows a group of friends who wake up from a wild night in Las Vegas and can't remember what happened.

User: Oh, I've already seen that one. Do you have any other recommendations?

Machine: How about "Bridesmaids"? It's a funny movie about a woman who becomes the maid of honor for her best friend's wedding and all the crazy events that ensue.

User: That sounds interesting. I'll give it a try. Thanks!

Conversation with a Machine modified by RLHF:

User: Hello, can you recommend a good movie to watch tonight?

Machine: Sure! What genre are you in the mood for?

User: I'm in the mood for a comedy.

Machine: How about "The Hangover"? It's a hilarious comedy that follows a group of friends who wake up from a wild night in Las Vegas and can't remember what happened.

User: Oh, I've already seen that one. Do you have any other recommendations?

Machine: I apologize for the repetition. Based on your preference for comedy, I suggest "Bridesmaids." It's a funny movie about a woman who becomes the maid of honor for her best friend's wedding and all the crazy events that ensue. It has received positive reviews and is known for its witty dialogue and comedic performances.

User: Thanks for the suggestion. Is there anything else you can recommend?

Machine: Certainly! If you're open to a different genre, I highly recommend "Inception." It's a mind-bending science fiction thriller that explores the concept of shared dreaming. The film is directed by Christopher Nolan and has garnered critical acclaim for its complex narrative and stunning visual effects.

User: Oh, I've heard about "Inception" before. It sounds intriguing. I think I'll give it a try. Thanks for the recommendation!

Non-RLHF vs RLHF

- **Human Feedback** in summerization

Prompt:
To pursue a Computer Sc.PhD or continue working? Especially if one has no real intention to work in academia even after grad school .

Vanilla LM:
I' m considering pursuing a PhD in Computer Science, but I'm worried about the future. Im currently employed full-time, but I'm worried about the future.

+

Human Annotation:
Software Engineer with a job I'm happy at (for now), deciding whether to pursue a PhD to improve quallfications and explore interests and a new challenge

=

RLHF Model:
Currently employed, considering pursuing PhD in Computer Science to avoid being stuck wih no residency visa ever again . Has anyone pursued a PhD purely for the sake of research, with no intention of joining the academic world?

RLHF - Optimizing on Human Preferences

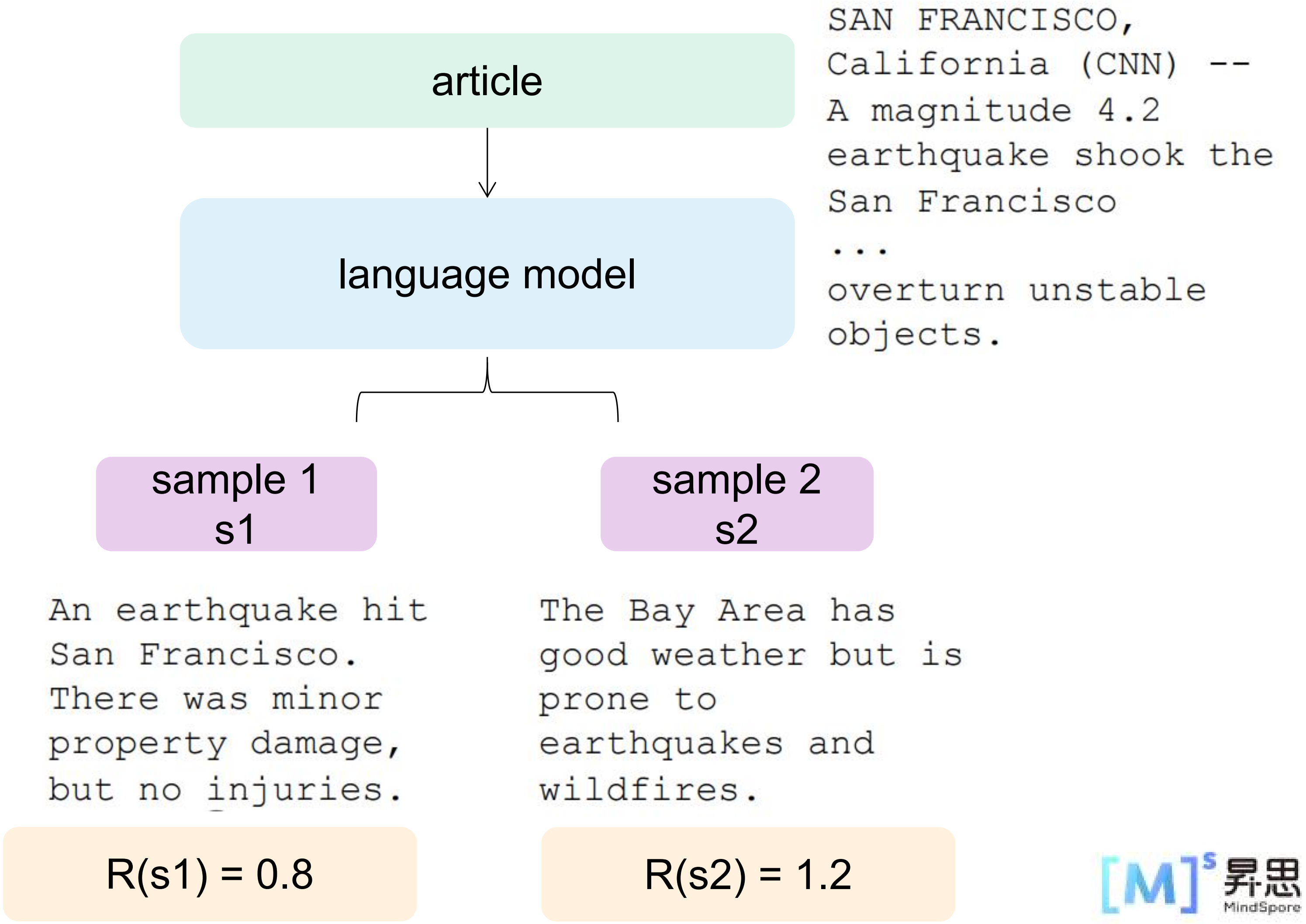
Example: Summarization Task

For each sample s, we assign a human reward R(s) to that summary.

The higher the reward, the better the summary is based on human preference.

The language model is optimized by maximizing the expected reward of samples.

$$\mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})] = \frac{1}{m}\sum_{1}^{M} R(s_i)$$



RL algorithm - Policy Gradient (PG)

- Supervised learning: use **gradient descent** method to **minimize the loss**
- Reinforcement learning: use **gradient ascent** method to **maximize the reward**
  - $\boldsymbol{\theta}$:parameter of model
  - $\boldsymbol{p_\theta(s_i)}$: probability of the ith language model sample
  - $\mathbf{m}$: number of iterations

$$\theta^{new} \leftarrow \theta^{old} + \eta \nabla \bar{R}_{\theta^{old}}$$

$$\nabla \overline{R_\theta} = \mathbb{E}_{\hat{s} \sim p_\theta(s)}[R(\hat{s})\,\nabla_\theta \log p_\theta(\hat{s})] \approx \frac{1}{m}\sum_{i=1}^{m} R(s_i)\,\nabla_\theta \log p_\theta(s_i)$$

The objective of reinforcement learning is to reinforce good actions by increasing the probability they happen again.

If R is +++ → **good action** → **take gradient steps to maximize $p_\theta(s_i)$**

If R is --- → **bad action** → **take gradient steps to minimize $p_\theta(s_i)$**

[M]^S 昇思 MindSpore

RL algorithm - Proximal Policy Optimization (PPO)

Policy Gradient → On-policy → Off-policy → Add constraint

- on-policy: The model being optimized and the model interacting with the human is **the same**.
  - **Have to recollect samples every time the model is updated**
- off-policy: The model being optimized and the model interacting with the human is **different**.
  - **No need to recollect samples**

on-policy to off-policy

$$\nabla\bar{R}_\theta = E_{\hat{s}\sim p_{\theta'}(s)}\left[\frac{p_\theta(s)}{p_{\theta'}(s)} R_{\theta'}(\hat{s}) \nabla log p_\theta(\hat{s})\right]$$

$$\bar{R}_\theta = E_{\hat{s}\sim p_{\theta'}(s)}\left[\frac{p_\theta(s)}{p_{\theta'}(s)} R_{\theta'}(\hat{s})\right]$$

Importance sampling: use a different distribution $p_{\theta'}(s)$ to model $p_\theta(s)$

add constraint

$$\bar{R}_\theta^{PPO} = \bar{R}_\theta - \beta KL(\theta, \theta')$$

KL divegence: penalize the divergence between θ, θ'

PPO2

$$\bar{R}_\theta^{PPO2} \approx \sum_{i} min\left(\frac{p_\theta(s)}{p_{\theta'}(s)} R_{\theta'}(s_i), clip\left(\frac{p_\theta(s)}{p_{\theta'}(s)}, 1-\varepsilon, 1+\varepsilon\right) R_{\theta'}(s_i)\right)$$

MindSpore 昇思

Human preference modelling

R(s): arbitrary, non-differentiable reward function

- **problem 1**: expensive human-in-loop
- **solution 1**: **model human preferences** as a separate NLP problem
- **problem 2**: noisy human feedbacks
- **solution 2**: use **pair-wise comparison** instead of direct scoring

```
An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.
```

s_1

$R(s_1) = 8.0$

```
The Bay Area has
good weather but is
prone to
earthquakes and
wildfires.
```

s_2

$R(s_2) = 1.2$

```
A 4.2 magnitude
earthquake hit
San Francisco,
resulting in
massive damage.
```

s_3

$R(s_3) = \quad 4.1? \quad 6.6? \quad 3.2?$

Human preference modelling

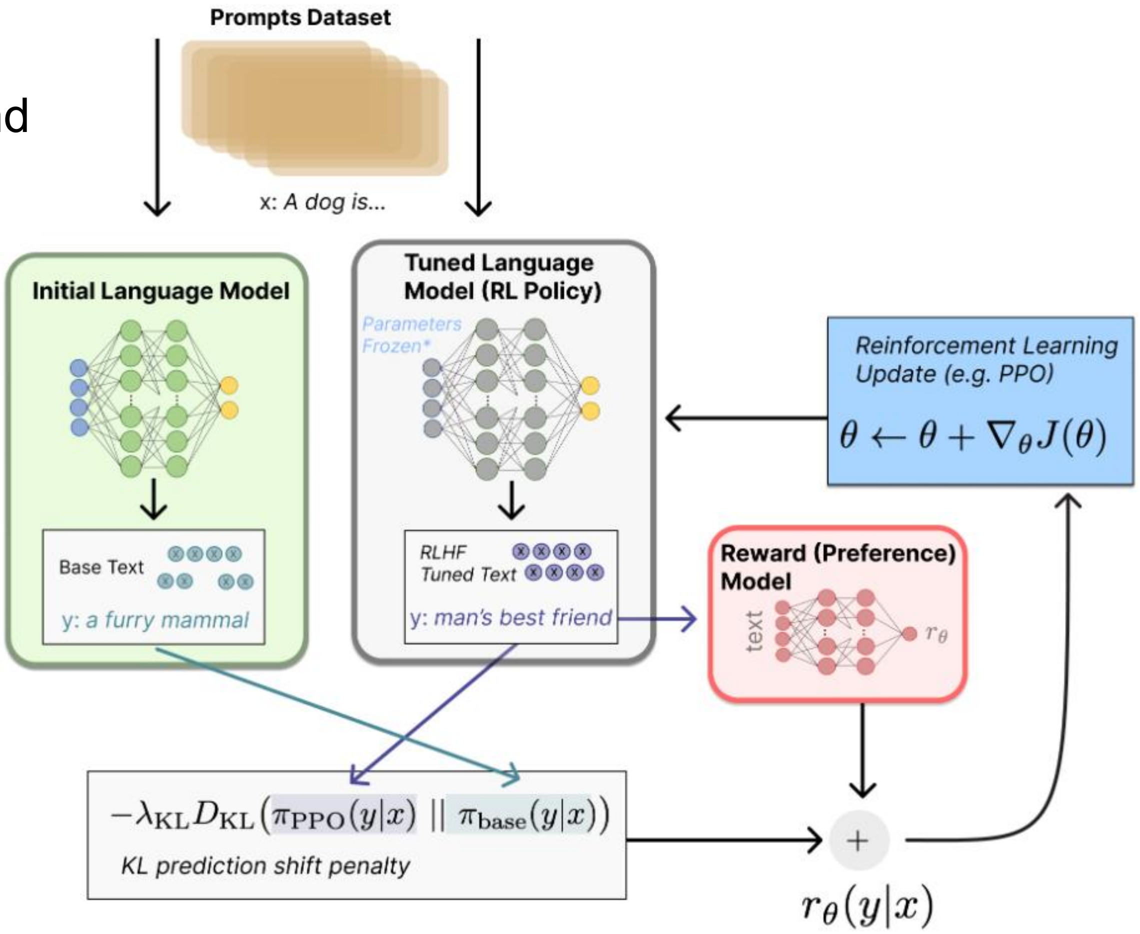
R(s): arbitrary, non-differentiable reward function

- problem 1: expensive human-in-loop
- solution 1: **model human preferences** as a separate NLP problem
- problem 2: noisy human feedbacks
- solution 2: ask for **pair-wise comparison** instead of direct scoring

An earthquake hit San Francisco. There was minor property damage, but no injuries.	>	A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.	>	The Bay Area has good weather but is prone to earthquakes and wildfires.
s_1		s_3		s_2

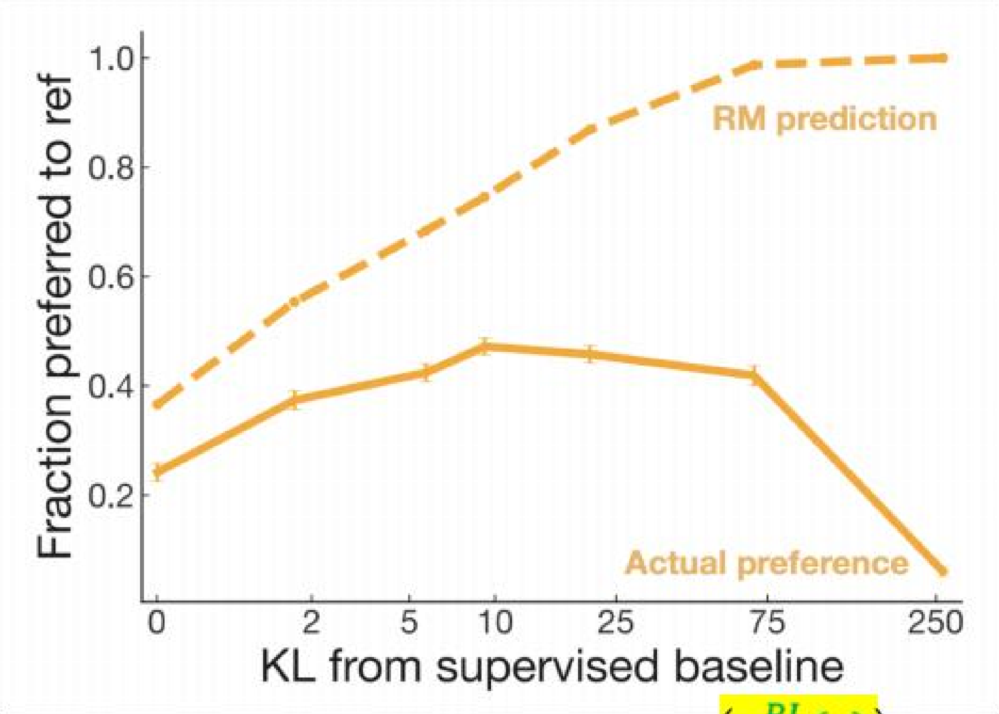
Review of RLHF

1. Pretraining a language model (LM),
2. gathering data and training a reward model, and
3. fine-tuning the LM with reinforcement learning.



Challenges with RLHF

- Human preferences are unreliable!
  - "Reward hacking" is a common problem in RL
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, ***regardless of truth***
  - This can result in making up facts + hallucinations
- **Models** of human preferences are *even more* unreliable!
- Consideration in dimension of both reward and time
  - does not investigate too much in **sequential process**
- Weakness in negotiation
  - ChatGPT does not do very well in **goal-directed dialogue**, where some messages might not be useful at the current moment but elict some information in the following conversation
  - rarely ask to clarify questions



$$R(s) = RM_{\phi}(s) - \beta \log\left(\frac{p_{\theta}^{RL}(s)}{p^{PT}(s)}\right)$$