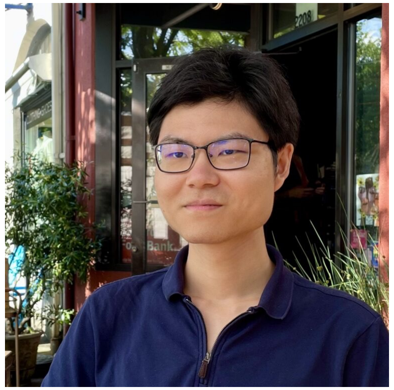
Reward Gaming in Conditional Text Generation

Richard Yuanzhe Pang[1] Vishakh Padmakumar[1] Thibault Sellam[2] Ankur P. Parikh[2] He He[1]

[1] NYU [2] Google DeepMind

July 2023





Conditional text generation

Machine translation

Data-to-text generation

Summarization

Question generation

Dialogue

Creative generation (stories, poems)

…

Input

夏威夷群岛低地的大部分降雨集中在冬季（十月至四月）。通常在5月到9月间比较干燥。热带风暴和偶尔的飓风通常发生在7月到11月之间。

Output

Most of the rainfall in the lowlands of the Hawaiian Islands is concentrated in winter (October to April). It is usually dry between May and September. Tropical storms and the occasional hurricane typically occur between July and November.

Goal: $\max_\theta \mathbb{E}_{\boldsymbol{y} \sim p_\theta}$ reward$(\boldsymbol{x}, \boldsymbol{y})$

RL is one possible algorithm

What's the reward?

- Summary saliency and faithfulness in Pasunuru and Bansal (2018)
- A summary scorer learned from human pairwise comparisons in Stiennon et al. (2020) and Wu et al. (2021; recursively summarizing books)
- An article-summary entailment classifier in Pang et al. (2021; agreement-oriented multi-doc summarization)
- BLEURT in Shu et al. (2021; reward optimization for NMT)

Motivating example: Increasing MT quality by expert feedback

Step 1: human annotation dataset D_{reward}

state-owned	enterprises	and			
1	1	1			

<u>advantageous</u>	private	enterprises
0	1	1

entered	the	<u>revolutionary</u>	<u>base</u>	<u>area</u>
1	1	0	0	0

Step 2: Train a reward function using D_{reward}

- *f* predicts whether each token is in a no-error span

Motivating example:
Increasing MT quality by expert feedback

Step 1: human annotation dataset D_{reward}

state-owned	enterprises	and			
1	1	1			
<u>advantageous</u>	private	enterprises			
0	1	1			
entered	the	<u>revolutionary</u>	<u>base</u>	<u>area</u>	
1	1	0	0	0	

Step 2: Train a reward function using D_{reward}

- f predicts whether each token is in a no-error span

Step 3: Train the sequence generation model using D_{task} by RL

- Reward going up ✓
- No improvement in BLEU; related: Shu et al. (2021)

<u>Example generations</u>

the 66 countries and regions have been able to <u>conduct</u> the evidence in the dissemination of the virus in 2015.

the newspaper in ankara has been able to <u>conduct</u> the military information and the military work in jordan and the disappearance of military work.

Motivating example:
Increasing MT quality by expert feedback

Step 1: human annotation dataset D_{reward}

state-owned enterprises and
1 1 1

advantageous private enterprises
0 1 1

entered the revolutionary base area
1 1 0 0 0

Step 2: Train a reward function using D_{reward}

- *f* predicts whether each token is in a no-error span

Step 3: Train the sequence generation model using D_{task} by RL

- Reward going up ✓
- No improvement in BLEU; related: Shu et al. (2021)

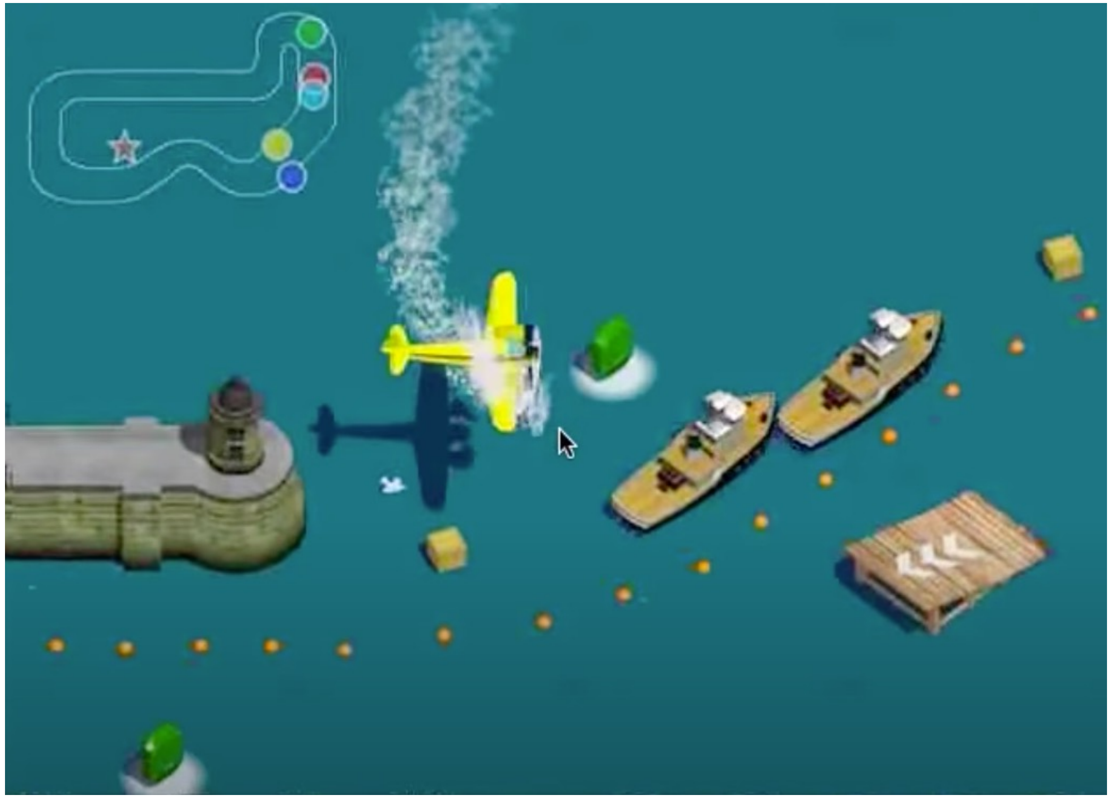
The model exploits the spurious correlation between “conduct” and high reward

Example generations

the 66 countries and regions have been able to conduct the evidence in the dissemination of the virus in 2015.

the newspaper in ankara has been able to conduct the military information and the military work in jordan and the disappearance of military work.

Lots of prior anecdotal evidence of reward gaming in gameplay / robotics



A classical example: the boat racing game

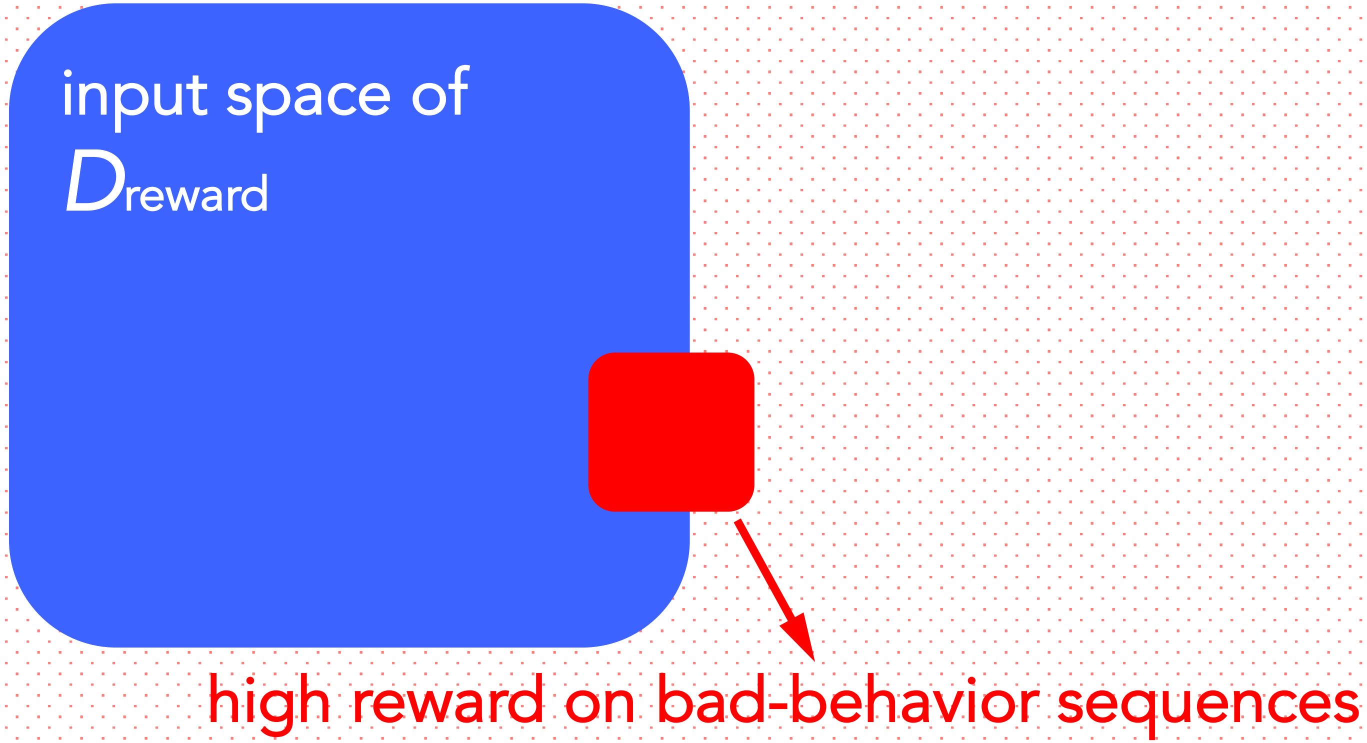
original goal: beat humans in boat racing
behavior: driving in circles & hitting things

reason: the reward is not designed to achieve the original intended goal

Image source: Jack Clark's YouTube upload
https://www.youtube.com/watch?v=tlOIHko8ySg

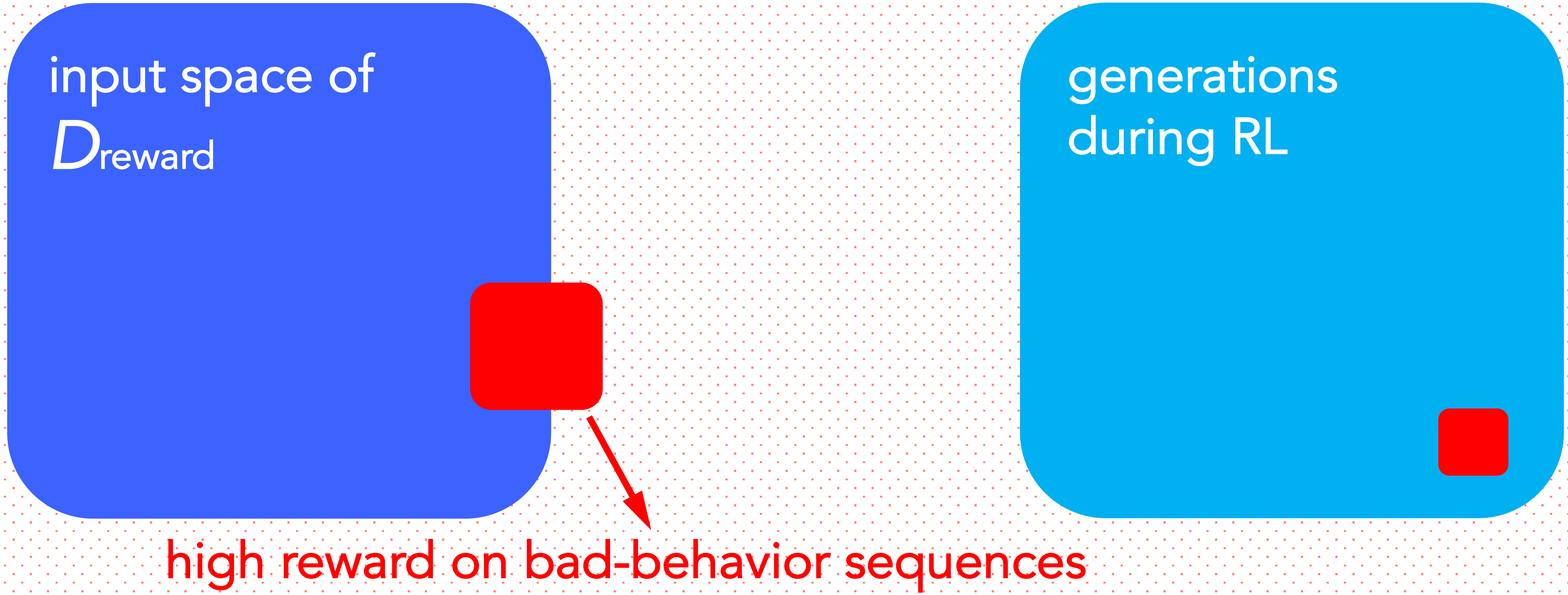
In conditional text generation, the rewards can be gamed

Undesirable patterns can get amplified during RL training of the generators



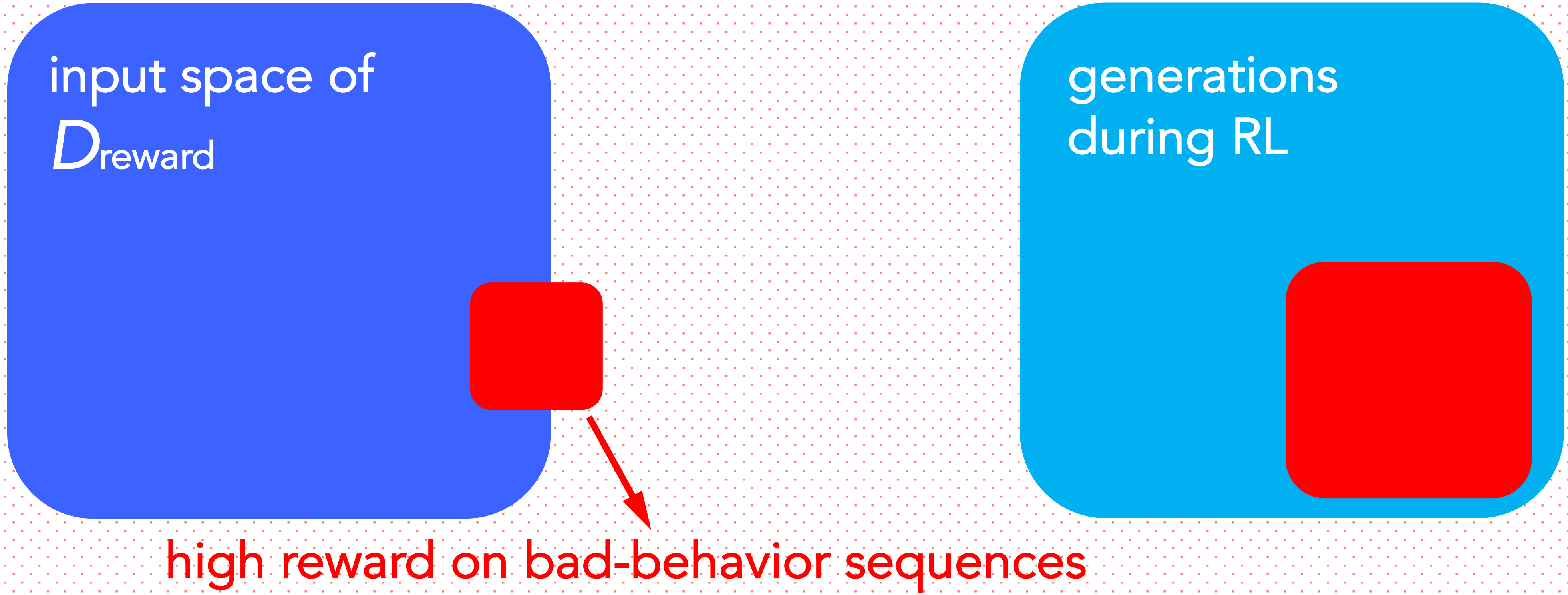
In conditional text generation, the rewards can be gamed

Undesirable patterns can get amplified during RL training of the generators



In conditional text generation, the rewards can be gamed

Undesirable patterns can get amplified during RL training of the generators



In conditional text generation, the rewards can be gamed

Undesirable patterns can get amplified during RL training of the generators

Three failure cases

A. Annotation errors
B. Naturally occurring spurious correlation
C. Underspecified behavior in reward

In conditional text generation, the rewards can be gamed

Undesirable patterns can get amplified during RL training of the generators

Three failure cases

A. Annotation errors

A group of examples could be misannotated systematically:

e.g., annotators carelessly labeling all long paragraphs as effective

In conditional text generation, the rewards can be gamed

Undesirable patterns can get amplified during RL training of the generators

Three failure cases

A. Annotation errors

A group of examples could be misannotated systematically:

e.g., annotators carelessly labeling all long paragraphs as effective

e.g., annotators carelessly label all generations with "according to Wikipedia" as truthful

In paper: even 0.05% annotation errors can lead to total generation failure

In conditional text generation, the rewards can be gamed

Undesirable patterns can get amplified during RL training of the generators

Three failure cases

A. Annotation errors e.g., a group of examples is misannotated systematically

B. Naturally occurring spurious correlation

e.g., short outputs tend to be more truthful – Lin et al. (2021)

e.g., the MT example discussed earlier

Increasing MT quality by expert feedback

Step 1: D_{reward}

state-owned	enterprises	and
1	1	1

advantageous	private	enterprises
0	1	1

entered	the	revolutionary	base	area
1	1	0	0	0

Step 2: Train a reward function using D_{reward}

- *f* predicts whether each token is in a no-error span

Step 3: Train the sequence generation model using D_{task} by RL

- Marginal improvement in BLEU; related: Shu et al. (2021)

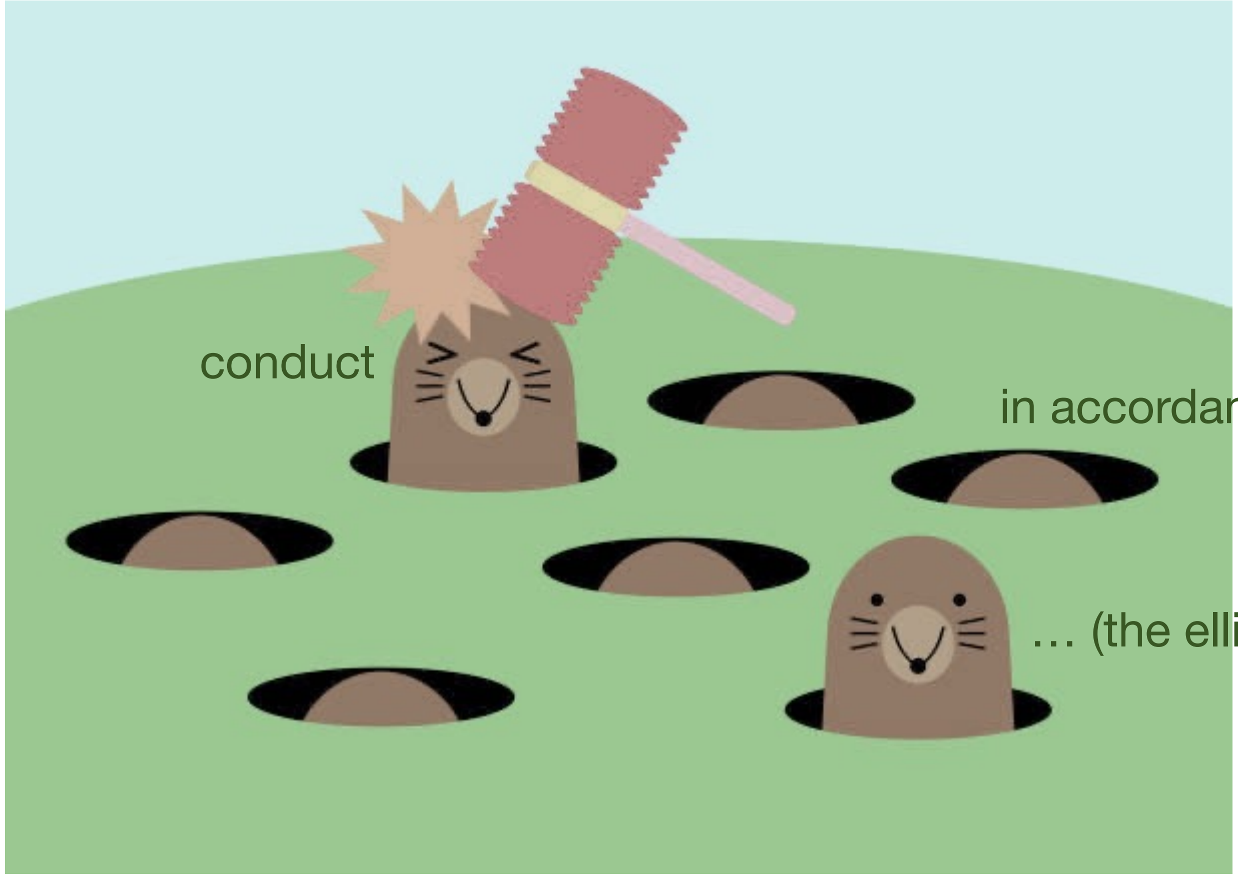
The model exploits the spurious correlation between "conduct" and high reward

Example generations

the 66 countries and regions have been able to conduct the evidence in the dissemination of the virus in 2015.

the newspaper in ankara has been able to conduct the military information and the military work in jordan and the disappearance of military work.

Can we just remove this spurious feature?



Credit: He He's recent talk (https://iclr.cc/virtual/2023/workshop/12827) & https://www.istockphoto.com/vector/a-game-to-hit-the-mole-gm1153033854-313029541

In conditional text gen, the rewards can (also) be gamed

Undesirable patterns can get amplified during RL training of the generators

Three failure cases

A. Annotation errors e.g., a group of examples is misannotated systematically

B. Naturally occurring spurious correlation e.g., short outputs tend to be more truthful; e.g., the MT example discussed earlier

C. Underspecified behavior in reward

e.g., dialogue agent trained to negotiate generates incomprehensible sentences, b/c those sentences are underspecified by the reward function (Lewis et al., 2017)

Potential remedies

Restricting the policy

- Regularizing toward the ML solution: interpolate RL & ML losses, interleave RL & ML updates, KL-regularized RL (popular)
    - It may be difficult to have the MLE-trained model
    - It may not always work, esp. when MLE-trained model is not good
    - Hard to tune coefficients for the KL term
- Restricting the policy by leveraging a discriminator (so the generations at least "look like" the sentences in a certain corpus)

Potential remedies

Fixing the reward itself

Iteratively collect human annotations

- Obtaining annotations -> training reward -> training generator -> more annotations -> training reward -> training generator -> etc.
- Concern: cost (e.g., MT MQM expert annotations are very expensive); hard to predict how many iterations we need

Takeaways

Using RL to train conditional text generation models **is not trivial**!

Reward gaming can happen (when high reward is assigned to bad behaviors); these bad behaviors can be amplified

- Annotation errors, naturally-occurring spurious correlation, underspecified behavior in reward

Open questions

- Effective ways to detect obscure gaming behavior in long generations
- Learning from feedback without RL?