Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Autores paper: Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning y Chelsea Finn.

Mayo 2023

Presentado por: Sebastián Burgos Martínez

python

Copy code

```python
def sum_of_first_n_even_numbers_method1(N):
    total = 0
    for i in range(1, N + 1):
        even_number = 2 * i
        total += even_number
    return total
```

python

Copy code

```python
def sum_of_first_n_even_numbers_method2(N):
    if N < 1:
        return 0
    return N * (N + 1)
```

Default (GPT-3.5)

Give me a Python function that calculates the sum of the first N even numbers

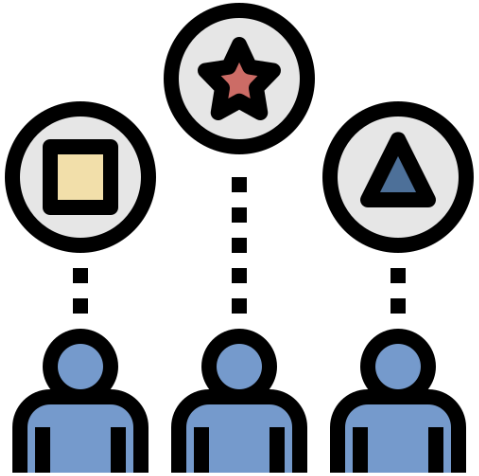
You can create a Python function to calculate the sum of the first N even numbers using a loop. Here's a simple example of such a function:

python Copy code

```python
def sum_of_first_n_even_numbers(N):
    if N <= 0:
        return 0  # If N is zero or negative, the sum is zero.

    even_sum = 0
    count = 0
    num = 2  # Start with the first even number, 2.

    while count < N:
        even_sum += num
        num += 2  # Increment to the next even number.
        count += 1

    return even_sum

# Example usage:
N = 5
result = sum_of_first_n_even_numbers(N)
print(f"The sum of the first {N} even numbers is: {result}")
```

Problema

- Entrenamiento con data generada por humanos con distintos objetivos, prioridades y competencias.
- Algunos de estos objetivos y competencias no son deseables de imitar.
- Queremos establecer preferencias en nuestro modelo.



Problema

- Código eficiente de alta calidad.
- Información correcta y certera.
- Generación de sentimientos (que la respuesta sea positiva y alentadora).

Contenidos

- Problema
- Trabajo previo
- Solución: Direct Preference Optimization (DPO)
- Resultados
- Discusión: conclusion y trabajo futuro

Trabajo previo

Reinforcement learning from human feedback (RLHF) pipeline[1]

1. Ziegler et al. 2020

Trabajo previo

Reinforcement learning from human feedback (RLHF) pipeline[1]

1) Supervised fine-tuning (SFT) phase

1) Collect demonstration data, and train a supervised policy

1. Ziegler et al. 2020

Trabajo previo

Reinforcement learning from human feedback (RLHF) pipeline[1]

1) Supervised fine-tuning (SFT) phase	1) Collect demonstration data, and train a supervised policy
2) Reward modelling phase	2) Collect comparison data, and train a reward model

1. Ziegler et al. 2020

Trabajo previo

Reinforcement learning from human feedback (RLHF) pipeline[1]

1) Supervised fine-tuning (SFT) phase

1) Collect demonstration data, and train a supervised policy

2) Reward modelling phase

2) Collect comparison data, and train a reward model

3) Reinforcement learning (RL) fine-tuning phase

3) Optimize a policy against the reward model using proximal policy optimization (PPO)

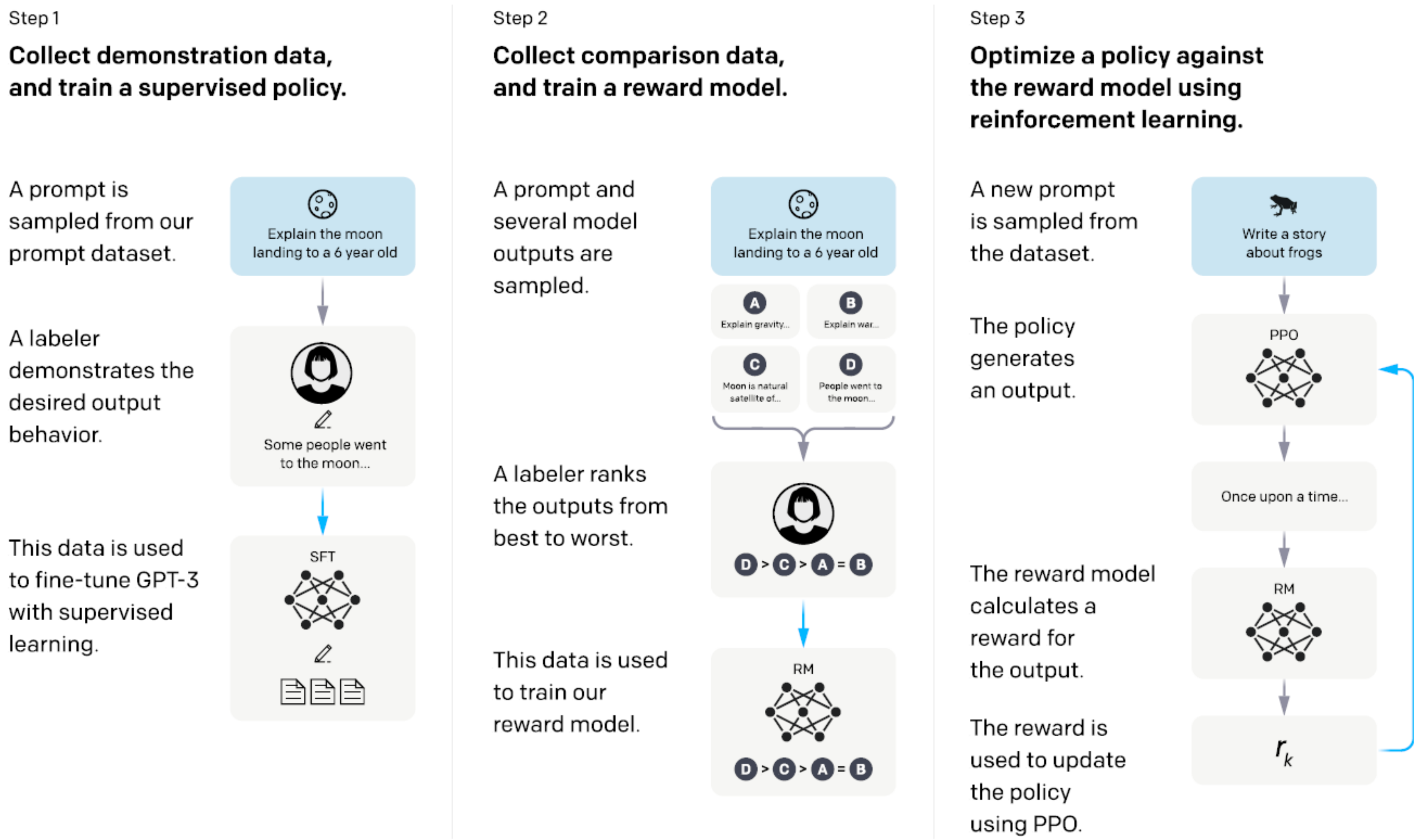
1. Ziegler et al. 2020

Trabajo previo

1) Supervised fine-tuning (SFT) phase

2) Reward modelling phase

3) Reinforcement learning (RL) fine-tuning phase



Solución: Direct Preference Optimization (DPO)

- Ocupa un enfoque sencillo para la optimización de la política.
- Evita el modelo de recompensa y el aprendizaje reforzado para optimizar directamente el modelo de lenguaje.
- Se pasa de una función de pérdida de recompensa a una función de pérdida sobre políticas.

Solución: Direct Preference Optimization (DPO)

RLHF: $$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}.$$

$$\Downarrow$$

DPO: $$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}$$

Solución: Direct Preference Optimization (DPO)

RLHF: $$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$$

$$\Downarrow$$

DPO: $$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

Solución: ¿qué hace DPO?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$

$$-\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \bigg[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}} \bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l} \bigg] \bigg]$$

Solución: ¿qué hace DPO?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$

$$- \beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \Bigg[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}} \Bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l} \Bigg] \Bigg]$$

$$\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

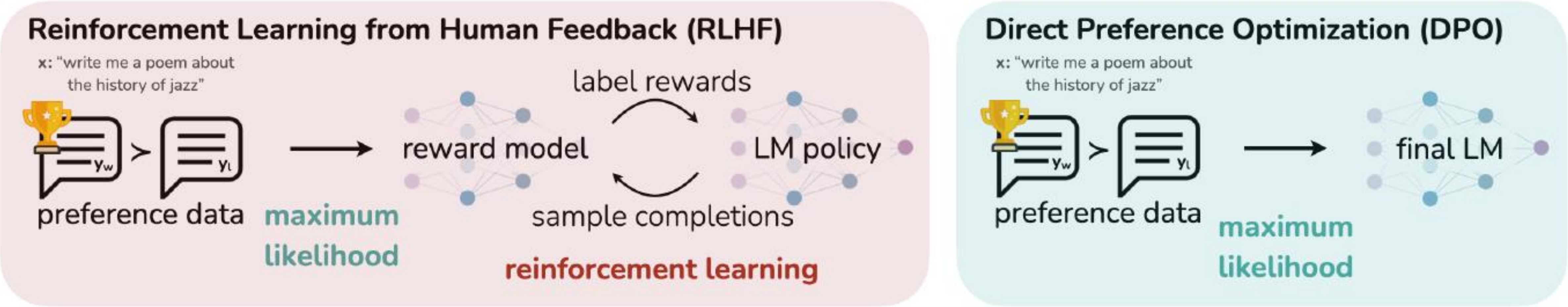
Solución: ¿qué hace DPO?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$
$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\underbrace{\sigma(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w))}_{\text{higher weight when reward estimate is wrong}}\Big[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Big]\Big]$$

$$\hat{r}_\theta(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \Longrightarrow$$

Your language model is secretly a reward model

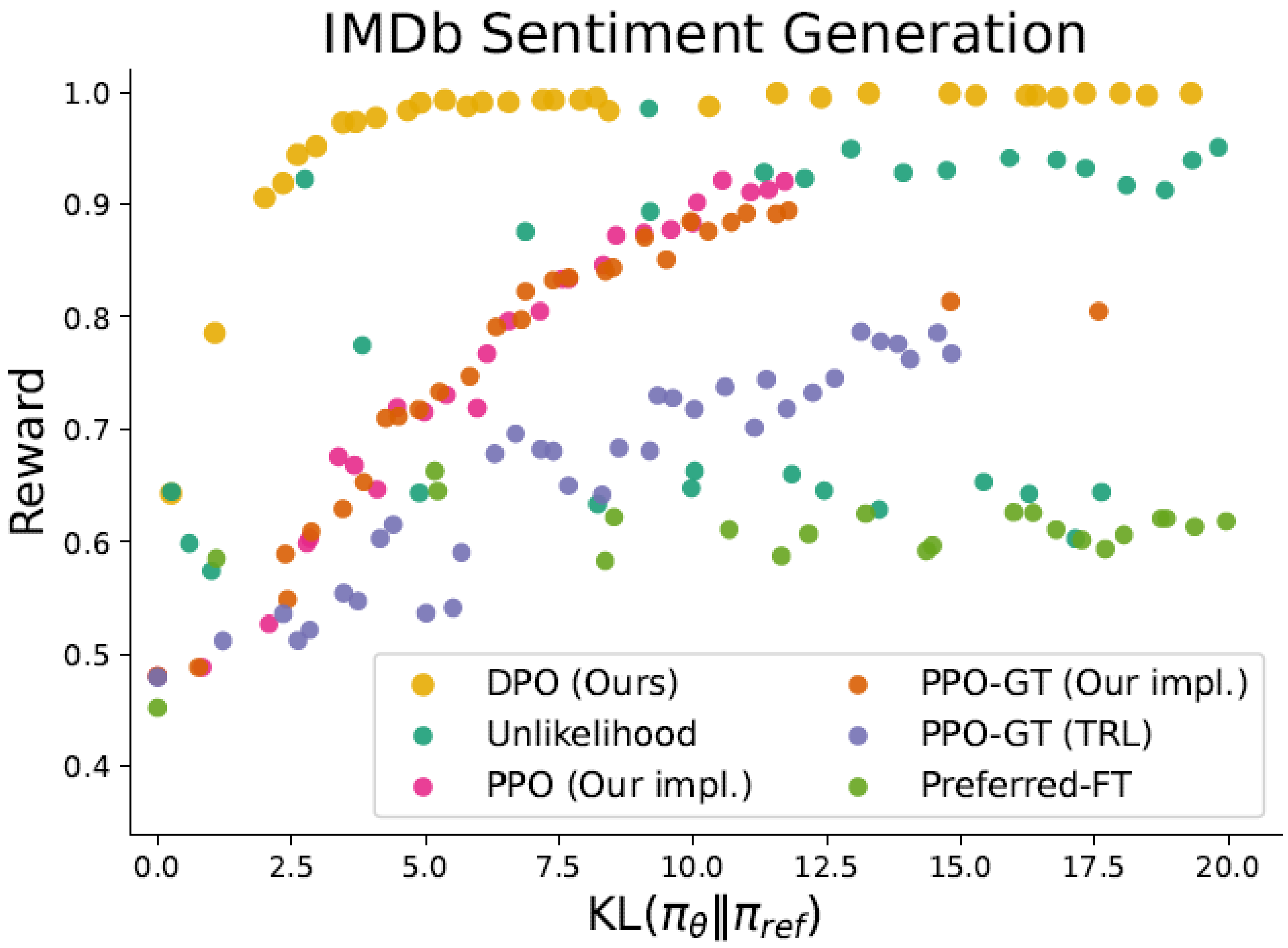
Solución: RLHF v/s DPO



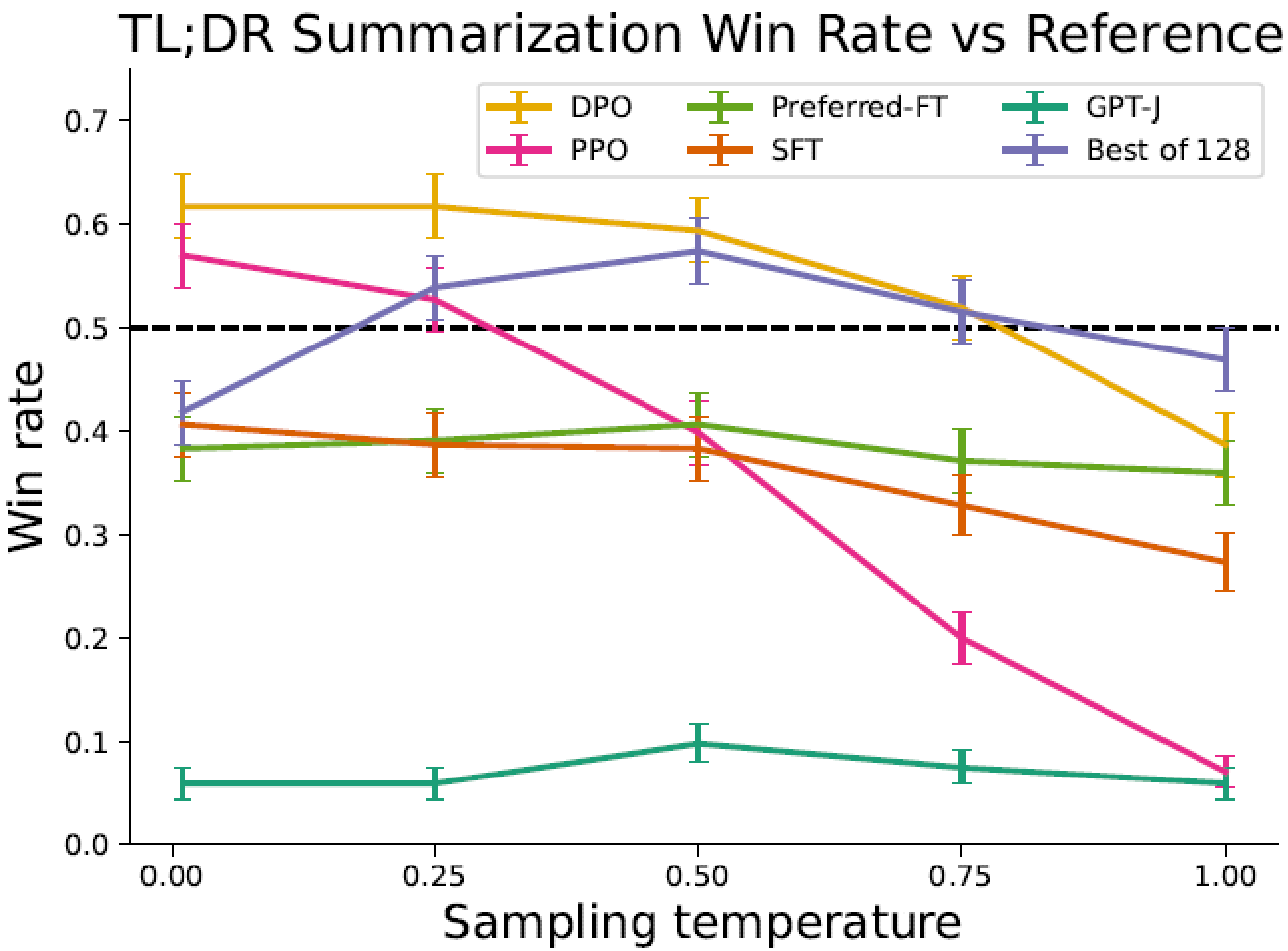
Resultados

- **Generación de sentimientos controlados:** IMBd dataset (fine-tuned GPT-2).
- **Resumen de texto:** Reddit TL;DR summarization dataset (fine-tuned SFT).
- **“Single-turn dialogue”:** Anthropic Helpful and Harmless dialogue dataset (fine-tuned off-the-shelf language model).

Resultados

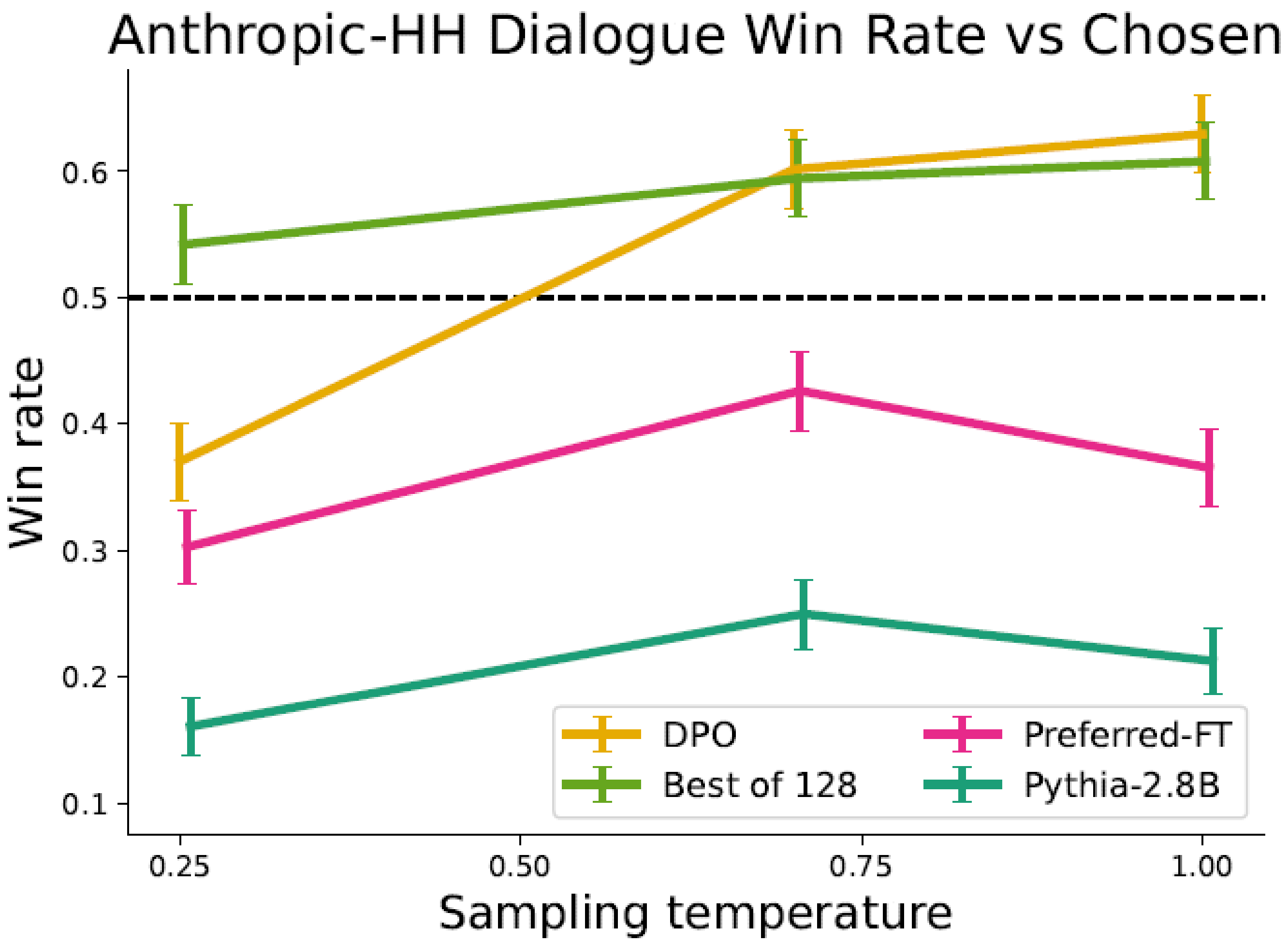

IMDb Sentiment Generation
Reward
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
$KL(\pi_\theta \| \pi_{ref})$
DPO (Ours)
Unlikelihood
PPO (Our impl.)
PPO-GT (Our impl.)
PPO-GT (TRL)
Preferred-FT

Resultados



TL;DR Summarization Win Rate vs Reference
DPO
PPO
Preferred-FT
SFT
GPT-J
Best of 128
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Win rate
0.00
0.25
0.50
0.75
1.00
Sampling temperature

Resultados



Anthropic-HH Dialogue Win Rate vs Chosen
Win rate
0.1
0.2
0.3
0.4
0.5
0.6
0.25
0.50
0.75
1.00
Sampling temperature
DPO
Best of 128
Preferred-FT
Pythia-2.8B

Discusión: conclusión

- DPO simplifica el entrenamiento de modelos con preferencias.
- DPO entrena directamente las preferencias con una simple cross-entropy loss, sin aprendizaje reforzado o pérdida de generalidad.
- DPO tiene un rendimiento similar o mejor que los algoritmos RLHF existentes (PPO).

Discusión: trabajo futuro

- ¿Cómo DPO rinde en otro tipo de tareas y en tareas distintas a las de entrenamiento, en comparación a los modelos de recompensa explícita?
- ¿Puede DPO entrenar con data no etiquetada haciendo un uso efectivo de esta?
- En la experimentación se ocuparon modelos con parámetros de hasta 6B, ¿cómo sería el rendimiento de DPO con modelos más grandes?

Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Autores paper: Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning y Chelsea Finn.

Mayo 2023

Presentado por: Sebastián Burgos Martínez