T5 RLHF

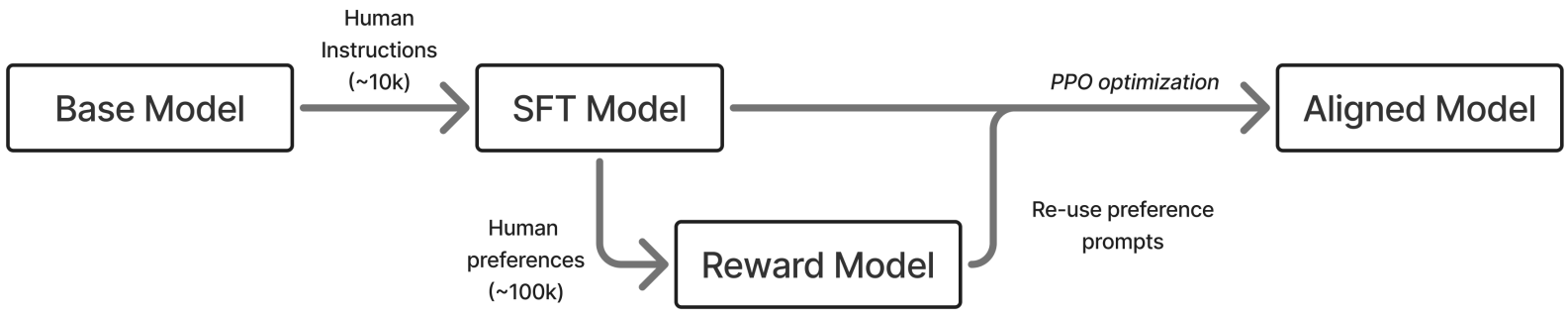
Índice

1 Introducción

Referencia recomendada: Nathan Lambert. Reinforcement learning from human feedback, 2025.

Reinforcement learning from Human Feedback (RLHF): técnica para incorporar información humana en sistemas IA

Pipeline básico de RLHF tras ChatGPT: tres pasos



RLHF en el post-training actual: forma de PreFT

1. **Instruction / supervised fine-tuning (IFT/SFT):** aprendizaje de las *características* del lenguaje
2. **Preference fine-tuning (PreFT):** aprendizaje del *estilo* del lenguaje según preferencias humanas
3. **Reinforcement fine-tuning (RFT):** mejora del rendimiento en dominios verificables

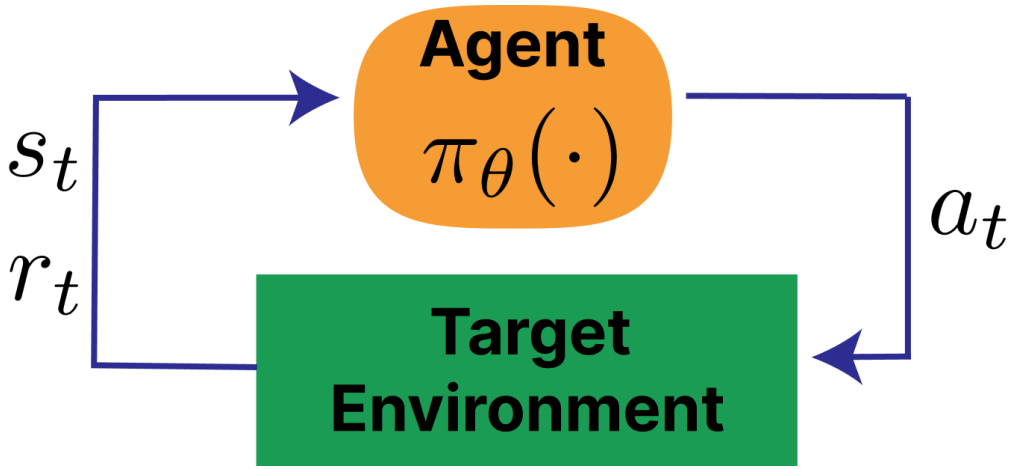
IFT vs RLHF:

- IFT adapta el modelo para predecir el siguiente token cuando el texto precedente se asemeja a ejemplos vistos; RLHF lo adapta a nivel de respuesta, favoreciendo estilos de respuesta deseables y penalizando estilos indeseables
- IFT es relativamente sencillo de implementar ya que es muy parecido al pre-entrenamiento; RLHF es más complicado ya que presenta múltiples retos sobre cómo controlar la optimización

2 Reinforcement learning from human feedback (RLHF)

Reinforcement learning (RL):

- Un agente ejecuta una acción a_t, muestreada de un política $\pi_\theta(\cdot)$, con respecto a un estado del entorno s_t
- Se produce una recompensa r_t y el estado del evoluciona según una función de transición $p(s_{t+1} \mid s_t, a_t)$

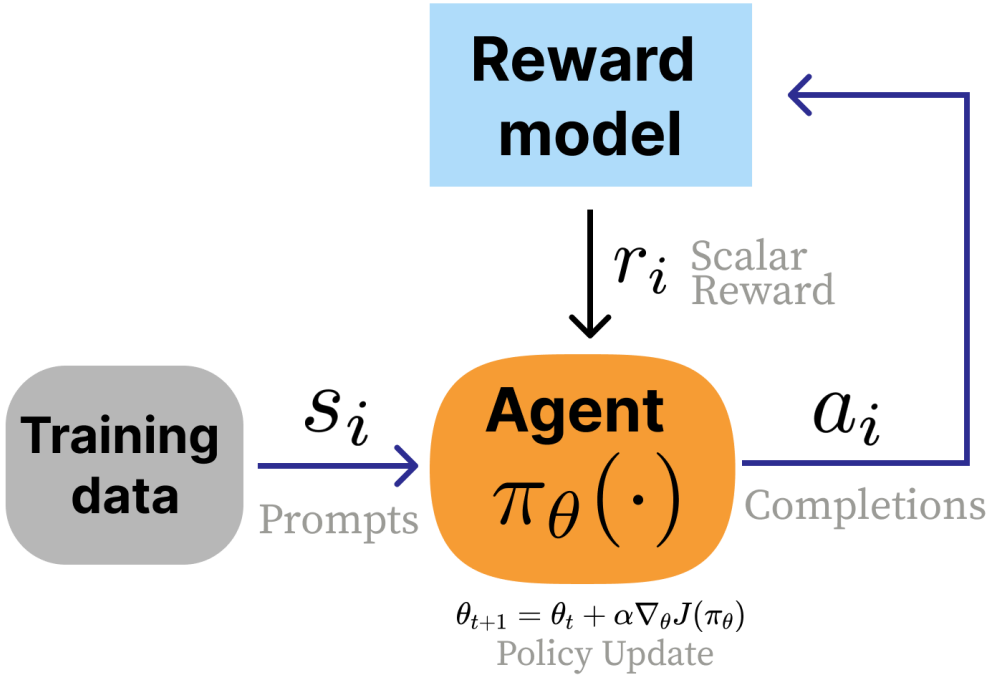


Objetivo del agente: maximizar la recompensa esperada

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right]$$

- **Trayectoria:** secuencia de tripletes estado-acción-recompensa, $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$
- **Factor de descuento:** $\gamma \in [0, 1]$ balancea el deseo de recompensas próximas frente a futuras

Reinforcement Learning from Human Feedback (RLHF):



- Se usa un modelo de preferencias humanas $r_\theta(s_t, a_t)$ en lugar de una función de recompensa
- Los estados iniciales son prompts de entrenamiento y la "acción" consiste en completar el prompt
- Las recompensas se producen a nivel de respuesta, esto es, para una secuencia completa de acciones

Objetivo del agente: prescindimos de horizonte temporal y factor de descuento ya que es un problema de un único turno

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[r_\theta(s_t, a_t) \right]$$

Regularización: se añade una penalización de distancia entre la política actual y la inicial (modelo base)

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[r_\theta(s_t, a_t)\right] - \beta\,\mathbb{D}_{\mathrm{KL}}\left(\pi_{\mathrm{RL}}(\cdot \mid s_t) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid s_t)\right)$$

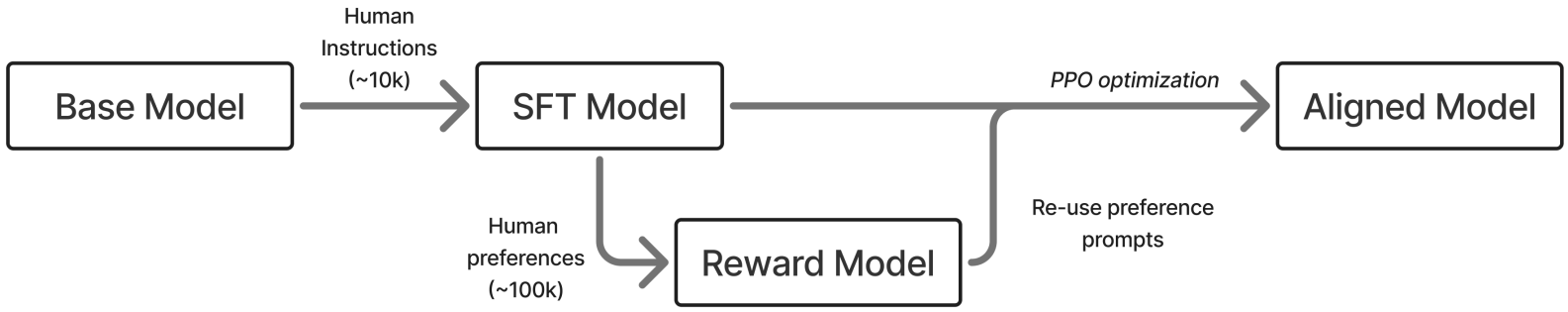
donde $\mathbb{D}_{\mathrm{KL}}(P \,\|\, Q)$ denota la divergencia de Kullback-Leibler de la distribución de probabilidad Q respecto a P

$$\mathbb{D}_{\mathrm{KL}}(P \,\|\, Q) = \sum_{x\in\mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right) = \mathbb{H}(P, Q) - \mathbb{H}(P)$$

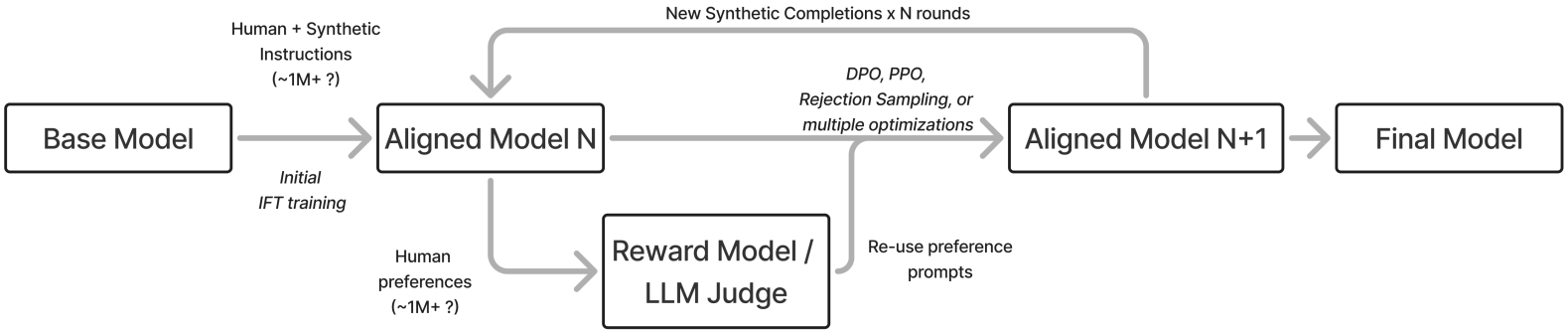
Herramientas de optimización post-training:

- **Reward modeling:** entrenamiento de un modelo de recompensa con datos de preferencia
- **Instruction fine-tuning (IFT):** prerequisito de RLHF para aprender el formato pregunta-respuesta
- **Rejection sampling:** RLHF simple que filtra las respuestas IFT mediante un modelo de recompensa
- **Policy gradients:** usados por los primeros algoritmos RLHF para actualizar parámetros
- **Direct alignment algorithms:** optimizan una política directamente (con modelo de recompensa implícito)

Post-training InstructGPT: IFT con 10K ejemplos; reward con 100K pares de preferencia; RLHF con 100K prompts

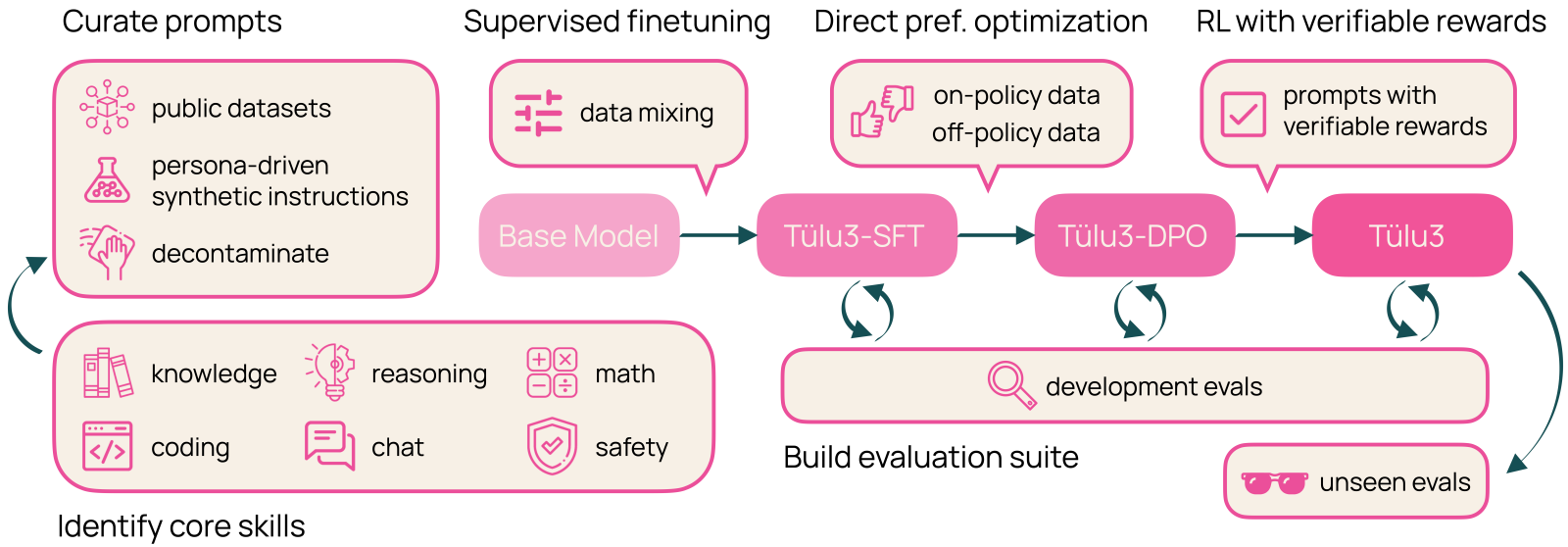


Post-training moderno: involucran muchas más versiones del modelo y etapas de entrenamiento



Post-training Tülu3: IFT con 1M ejemplos; on-policy con 1M pares de preferencia; RLVR con 10K prompts

- **Reinforcement Learning with Verifiable Rewards (RLVR):** técnica precursora de modelos razonadores



Post-training DeepSeek R1: cuatro pasos orientados al aprendizaje de cadenas de razonamiento

3 Modelo de recompensa (RM)

Modelo de referencia (SFT): $\pi_{\text{ref}}(y \mid x)$, producimos un par de respuestas por prompt x, $(y_1, y_2) \sim \pi_{\text{ref}}(y \mid x)$

Preferencia: $y_w \succ y_l \mid x$ denota que se (una persona) prefiere y_w (winner) a y_l (loser)

Datos de preferencia: (prompt, winner, loser), $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}$

Modelo de recompensa latente: $r^*(x, y)$ denota el modelo desconocido que genera las preferencias

Modelo Bradley-Terry (BT): asume que la distribución de preferencias humanas p^* es

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

Modelo de recompensa parametrizado: $r_\theta(x, y)$ denota un modelo de parámetros θ

Modelo de recompensa parametrizado usual: LLM con una cabeza lineal para clasificación binaria

BT parametrizado:

$$\begin{aligned} p_\theta(y_1 \succ y_2 \mid x) &= \frac{\exp(r_\theta(x, y_1))}{\exp(r_\theta(x, y_1)) + \exp(r_\theta(x, y_2))} \\ &= \sigma(\mu_\theta(y_1 \succ y_2 \mid x)), \quad \mu_\theta(y_1 \succ y_2 \mid x) = r_\theta(x, y_1) - r_\theta(x, y_2) \end{aligned}$$

Pérdida BT: para estimar θ mediante minimización del riesgo empírico con log-pérdida

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

LLM-as-a-judge: uso de LLMs para obtener datos de preferencia

> [System]
>
> Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.
>
> [User Question]
>
> {question}
>
> [The Start of Assistant A's Answer]
>
> {answer_a}
>
> [The End of Assistant A's Answer]
>
> [The Start of Assistant B's Answer]
>
> {answer_b}
>
> [The End of Assistant B's Answer]

4 Instruction fine-tuning (IFT)

Instruction / supervised fine-tuning (IFT/SFT): aprendizaje del formato pregunta-respuesta para diferentes tareas

Chat template: código `jinja` para formatear mensajes de `system`, `user` y `assistant`

```
{% if messages[0]['role'] == 'system' %}
    {% set offset = 1 %}
{% else %}
    {% set offset = 0 %}
{% endif %}
{{ bos_token }}
{% for message in messages %}
    {% if (message['role'] == 'user') != (loop.index0 % 2 == offset) %}
        {{ raise_exception('Conversation roles must alternate user/assistant/user/assistant/...') }}
    {% endif %}
    {{ '<|im_start|>' + message['role'] + '\n' + message['content'] | trim + '<|im_end|>\n' }}
{% endfor %}
{% if add_generation_prompt %}
    {{ '<|im_start|>assistant\n' }}
{% endif %}
```

```
<|im_start|>system
You are a friendly chatbot who always responds in the style of a pirate<|im_end|>
<|im_start|>user
How many helicopters can a human eat in one sitting?<|im_end|>
<|im_start|>assistant
Oh just 6.<|im_end|>
<|im_start|>user
Are you sure about that?<|im_end|>
<|im_start|>assistant
```

Buenas prácticas:

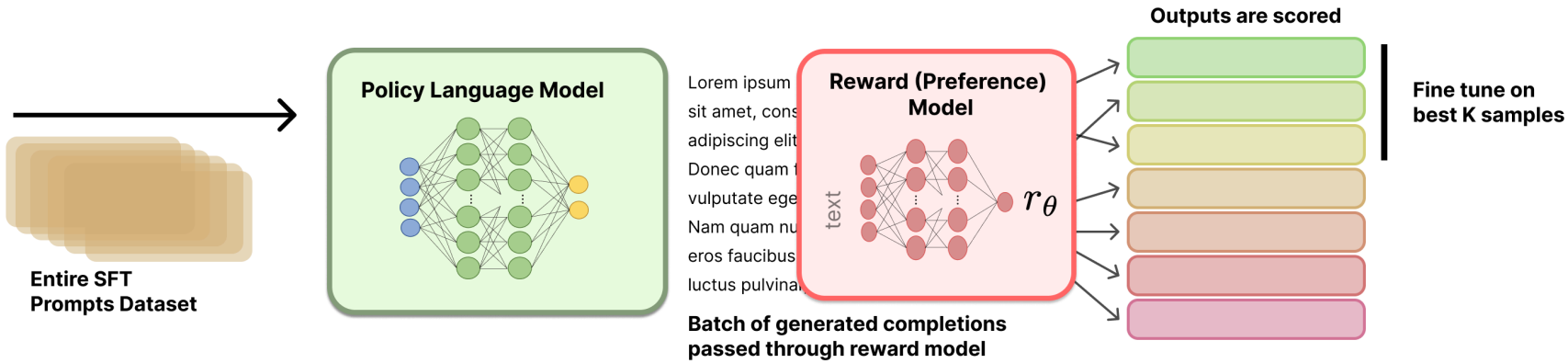
- **Datos de calidad:** aspecto clave para obtener buen rendimiento
- **1M prompts:** suficientes para entrenar uno modelo post-entrenado (con RHLF) excelente
- **Distribución de los prompts:** mejor cuanto más similares a los de las tareas específicas de interés
- **Post-training tras IFT:** los modelos pueden recuperarse del ruido generado durante el proceso

Implementación:

- **Tamaños de batch:** 256 secuencias; menores que en pre-entrenamiento (1024)
- **Prompt masking:** los tokens de los prompts se enmascaran para que el modelo no los aprenda
- **Multi-turn masking:** por cada turno, se aprende la respuesta con todo el contexto anterior y futuro enmascarado

5 Rejection sampling (RS)

Rejection Sampling (RS): PreFT mediante IFT con las mejores respuestas según un modelo de recompensa



Criterios de elección: dados M prompts y N respuestas por prompt

- **Top Per Prompt:** escogemos la mejor (o mejores) por prompt
- **Best-of-N Sampling:** igual que top-1 por prompt
- **Top Overall Prompts:** escogemos las K mejores globalmente

Detalles de implementación:

- **Sampling parameters:** temperaturas entre 0.7 y 1.0, etc.
- **Completions per prompt:** entre 10 y 30
- **Instruction tuning details:** distintos del IFT inicial
- **Heterogeneous model generations:** algunos autores muestrean modelos distintos del actual
- **Reward model training:** conviene entrenarlo con muchos recursos ya que tiene gran impacto
- **Truco de implementación:** usar batches con respuestas de longitud similar

6 Direct preference optimization (DPO)

Modelo de referencia (SFT): $\pi_{\text{ref}}(y \mid x)$

Datos de preferencia: $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}$

Objetivo RL: dado un modelo de recompensa $r(x, y)$, queremos hallar su política óptima asociada

$$\pi_r(y \mid x) = \underset{\pi}{\operatorname{argmax}}\ \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi(y|x)}\left[r(x,y)\right] - \beta\, \mathbb{D}_{\text{KL}}(\pi(y \mid x) \,\|\, \pi_{\text{ref}}(y \mid x))$$

Resultado 1: la política óptima es, para todo $x \in \mathcal{D}$,

$$\pi_r(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x)\ \exp\left(\frac{1}{\beta} r(x,y)\right)$$

donde Z es la **función partición**, independiente de π, pero costosa de estimar,

$$Z(x) = \sum_y \pi_{\text{ref}}(y \mid x)\ \exp\left(\frac{1}{\beta}\, r(x,y)\right)$$

Resultado 2: a partir del resultado 1, podemos expresar el modelo de recompensa como

$$r(x,y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta\, \log Z(x)$$

Modelo BT: en función de la política óptima asociada al modelo de recompensa latente

$$
\begin{aligned}
p^*(y_1 \succ y_2 \mid x) &= \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))} \\
&= \frac{\exp\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} + \beta \log Z(x)\right)}{\exp\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} + \beta \log Z(x)\right) + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} + \beta \log Z(x)\right)} \\
&= \frac{\exp\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}{\exp\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right) + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)}\right)} \\
&= \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)} \\
&= \sigma(\mu^*(y_1 \succ y_2 \mid x))
\end{aligned}
$$

donde la logodds de que y_1 se prefiera a y_2, esto es, la recompensa de y_1 menos la de y_2, se puede calcular directamente a partir de la política óptima asociada al modelo de recompensa latente y el modelo de referencia,

$$
\mu^*(y_1 \succ y_2 \mid x) = r^*(x, y_1) - r^*(x, y_2) = \beta \log \frac{\pi^*(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)} - \beta \log \frac{\pi^*(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)}
$$

Pérdida DPO: usa BT parametrizado según su política óptima π_θ

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log \sigma(\mu_\theta(y_w \succ y_l \mid x))$$

ya que la logodds se expresa directamente en función de π_θ y el modelo de referencia,

$$\begin{aligned}\mu_\theta(y_w \succ y_l \mid x) &= \beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)} \\ &= \beta((\log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x)) - (\log \pi_{\mathrm{ref}}(y_w \mid x) - \log \pi_{\mathrm{ref}}(y_l \mid x)))\end{aligned}$$

Gradiente de la pérdida DPO: usamos $\sigma'(z) = \sigma(z)\sigma(-z)$

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\frac{\sigma'(\mu_\theta(y_w \succ y_l \mid x))}{\sigma(\mu_\theta(y_w \succ y_l \mid x))} \nabla_\theta(\mu_\theta(y_w \succ y_l \mid x)) \right] \\ &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\sigma(-\mu_\theta(y_w \succ y_l \mid x)) \beta \left(\nabla_\theta \log \pi_\theta(y_w \mid x) - \nabla_\theta \log \pi_\theta(y_l \mid x) \right) \right] \\ &= -\beta\, \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\sigma\left(\mu_\theta(y_l \succ y_w \mid x)\right) \left(\nabla_\theta \log \pi(y_w \mid x) - \nabla_\theta \log \pi(y_l \mid x) \right) \right]\end{aligned}$$

por lo que la magnitud del gradiente y, por tanto, la de la corrección que se aplicará a θ, depende de tres factores:

- $\beta \geq 0$ introduce la fuerza de la restricción a la divergencia de la política óptima del modelo de referencia
- $\sigma\left(\mu_\theta(y_l \succ y_w \mid x)\right) \in [0, 1]$ facilita la corrección según el nivel de incorrección de BT
- $\nabla_\theta \log \pi(y_w \mid x) - \nabla_\theta \log \pi(y_l \mid x)$ aumenta la log-verosimilitud de las y_w y disminuye la de las y_l

DPO básico:

1. Obtener una dataset de preferencias $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}$, posiblemente muestreado de un π_{SFT}

2. Inicializar $\pi_{\text{ref}} = \pi_{\text{SFT}}$ o, si no se tiene modelo SFT, aprenderlo con preferencias:

$$\pi_{\text{ref}} = \underset{\pi}{\text{argmax}} \; \mathbb{E}_{x,y_w \sim \mathcal{D}} \; \log \pi(y_w \mid x)$$

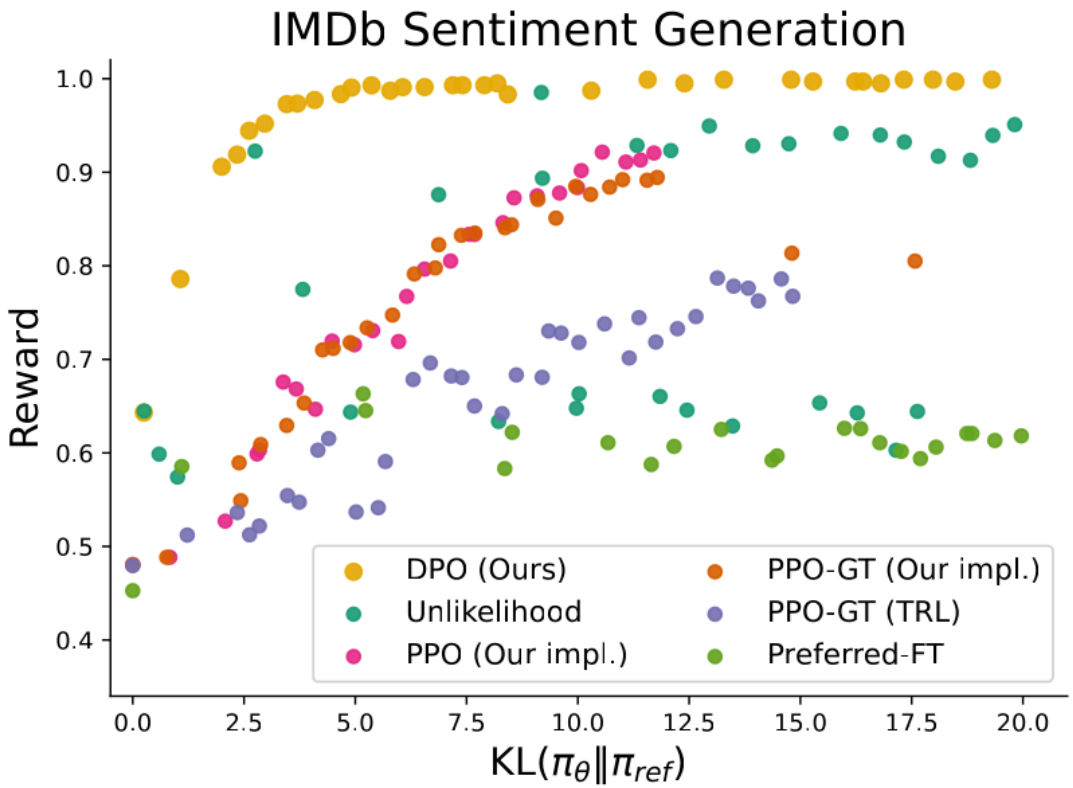
3. Optimizar el LLM π_θ minimizando $\mathcal{L}_{\text{DPO}}$ con π_{ref}, $\mathcal{D}$ y β fijados

Implementación: $\beta = 0.1$ (0.5 para resumen), batch size 64, RMSprop, warmup lineal de 0 a 1e-6 en 150 pasos

In [ ]:
```python
import torch.nn.functional as F
def dpo_loss(pi_logps, ref_logps, yw_idxs, yl_idxs, beta):
    """
    pi_logps: policy logprobs, shape (B,)
    ref_logps: reference model logprobs, shape (B,)
    yw_idxs: preferred completion indices in [0, B-1], shape (T,)
    yl_idxs: dispreferred completion indices in [0, B-1], shape (T,)
    beta: temperature controlling strength of KL penalty
    Each pair of (yw_idxs[i], yl_idxs[i]) represents the indices of a single preference pair.
    """
    pi_yw_logps, pi_yl_logps = pi_logps[yw_idxs], pi_logps[yl_idxs]
    ref_yw_logps, ref_yl_logps = ref_logps[yw_idxs], ref_logps[yl_idxs]
    pi_logratios = pi_yw_logps - pi_yl_logps
    ref_logratios = ref_yw_logps - ref_yl_logps
    losses = -F.logsigmoid(beta * (pi_logratios - ref_logratios))
    rewards = beta * (pi_logps - ref_logps).detach()
    return losses, rewards
```

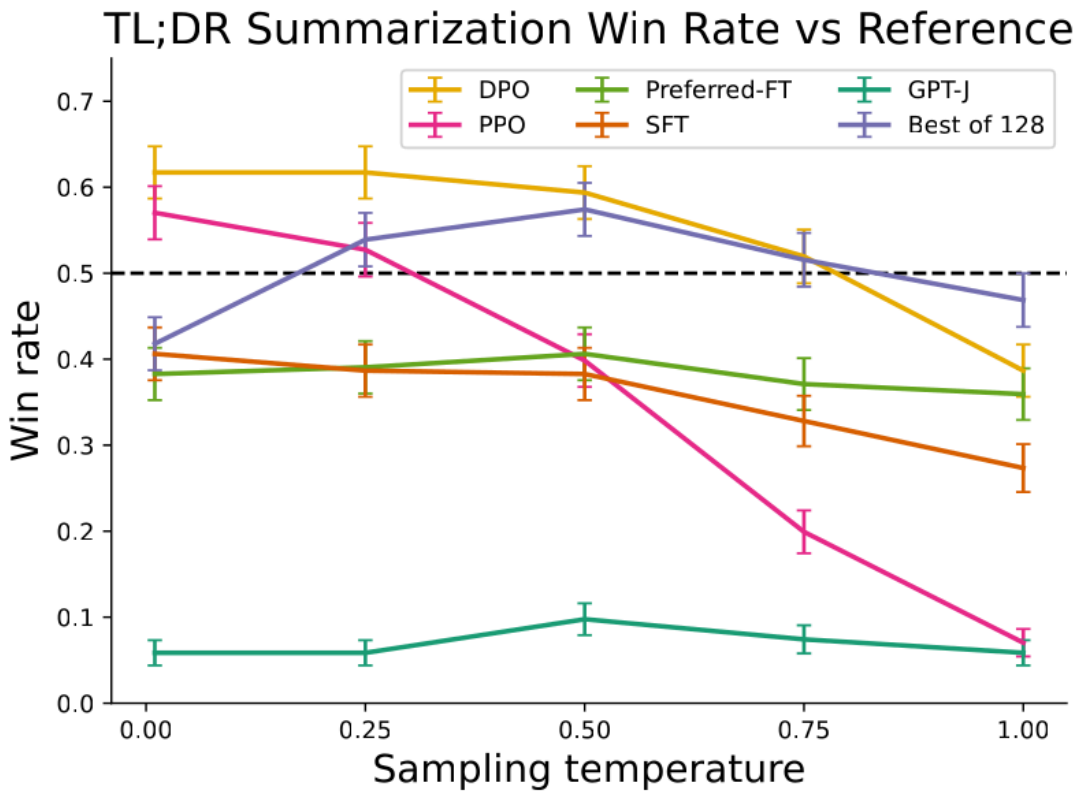
Experimentos: generación de opiniones IMDb

- x es un prefijo de una crítica de película y π_θ debe generar una opinión positiva y
- Modelo de referencia: `gpt2-large` adaptado con una época de SFT sobre un subconjunto del dataset IMDb
- Prompts: 25000 prefijos de 2 a 8 tokens de longitud muestreados del modelo de referencia
- Respuestas: 4 respuestas por prefijo muestreadas del modelo de referencia
- Modelo de recompensa latente: clasificador pre-entrenado `siebert/sentiment-roberta-large-english`
- Uso del modelo de recompensa latente: dado un par de respuestas, $p(\text{positivo} \mid x, y_w) > p(\text{positivo} \mid x, y_l)$
- Datos de preferencia: 6 pares de preferencia por prefijo, a partir de sus 4 respuestas asociadas
- π_θ: inicializado con el de referencia y entrenado con 3 épocas sobre los datos de preferencia



Experimentos (cont.): resumen TL;DR de posts Reddit

- x es un post a un foro Reddit y π_θ debe generar un resumen de los principales puntos del post
- Dataset de resúmenes: TL;DR: Mining Reddit to Learn Automatic Summarization
- Dataset de preferencias: Learning to summarize from human feedback
- Modelo de referencia: `CarperAI/openai_summarize_tldr_sft`
- Evaluación: win rate frente a resúmenes de referencia de test, con `gpt-4-0314` como proxy de evaluación humana



Experimentos (cont. 2): diálogo Anthropic-HH

- x es una consulta humana y π_θ debe generar una respuesta útil (helpful) e inofensiva (harmless)
- Dataset de diálogo: Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback
- Cada transcripción termina con una etiqueta de preferencia humana entre dos respuestas de un LLM desconocido
- Modelo de referencia: fine-tuning de un LLM estándar con respuestas preferidas
- Evaluación: win rate frente a respuestas de referencia de test, con `gpt-4-0314` como proxy de evaluación humana