Aligning KoGPT2 for Korean Dialogue: A Comparative Study of SFT–RM–PPO and SFT–DPO Pipelines

Han-gyeol Lim
Aiffel Research Course
https://github.com/gksruf293/AIFFEL_quest_rs

***Abstract*—Large language models (LLMs) such as ChatGPT show impressive fluency, yet they are primarily trained on English, yielding sub-optimal performance in Korean. We upgrade KoGPT2—a lightweight Korean GPT-2 variant—via two human-feedback pipelines: *(i)* the conventional SFT → Reward Model → PPO route and *(ii)* the recent SFT → DPO alternative. Using identical supervised and preference data, we detail each method and provide the first quantitative comparison on Korean dialogue generation.**

***Index Terms*—KoGPT2, RLHF, PPO, DPO, Korean chatbot**

I. Introduction

Large language models (LLMs) such as ChatGPT and GPT-4 have demonstrated remarkable conversational competence and broad reasoning skills. Nevertheless, because their pre-training corpora are dominated by English data, performance in low-resource languages like Korean remains sub-optimal. KoGPT2—a lightweight Korean GPT-2 derivative—offers fast inference yet struggles with nuanced instructions and factual grounding due to its limited native corpus.

Reinforcement Learning from Human Feedback (RLHF) is the de-facto method for aligning LLMs with human preferences. In its canonical form— **Supervised Fine-Tuning (SFT) → Reward Model (RM) → Proximal Policy Optimisation (PPO)**—the workflow introduces additional networks beyond the main policy:

- **Policy / Actor**generates responses and is updated by PPO.
- **Critic / Value head**estimates $V(s)$ for advantage calculation.
- **Reward Model**regresses a scalar reward from human-rated pairs; its parameters are usually frozen during PPO.
- **Reference Model**a static copy of the SFT checkpoint that anchors the KL penalty.

Open-source implementations (e.g., TRL's `PPOTrainer`) typically keep these components as *four separate parameter sets*. Memory- or speed- optimised variants may merge, for example, the actor and critic trunks or compute the KL penalty on-the-fly, yet practitioners still juggle up to four logical models, each with its own optimiser state and checkpoint. This proliferation complicates hyper-parameter tuning, slows iteration, and inflates deployment artefacts.

A recent alternative, *Direct Preference Optimisation* (DPO), collapses the entire stack into *one trainable policy*. Instead of learning an explicit reward signal, DPO directly maximises the likelihood that the policy assigns higher probability to a preferred answer than to a rejected one for the same prompt. By eliminating auxiliary networks and their interaction loops, DPO unifies data loading, simplifies distributed training, and eases deployment—reducing engineering overhead while maintaining, or even improving, alignment quality.

Contributions.: We make three main contributions:

- Construct a comprehensive Korean RLHF dataset that combines supervised examples and pairwise preference annotations from diverse sources.
- Align KoGPT2 using both the traditional **SFT–RM–PPO** pipeline and the structurally simpler **SFT–DPO** alternative under identical conditions.
- Provide the first quantitative comparison of these pipelines on Korean dialogue generation, showing that DPO matches or surpasses PPO while *dramatically simplifying* the training and deployment stack.

II. Dataset Construction

Our alignment workflow consists of a *shared* supervised stage (SFT), after which it branches into two alternative pipelines:

Pipeline A (SFT → Reward Model → PPO)
Full RLHF route that trains an explicit reward model and uses PPO.

Pipeline B (SFT → DPO)
Direct Preference Optimisation that skips reward modelling.

A. Question Harvest

We collect 167 577 Korean prompts from four corpora—ChatbotData (11 824), AI Hub Dialogue (49 711), AI Hub Commonsense (100 268) and KorQuAD (5 774)—then sample 12 000 diverse prompts spanning chit-chat to encyclopedic queries.

B. Supervised Fine-Tuning Data (shared)

- **Single-turn QA (12000)** – ChatGPT answers each prompt forming {`prompt, completion`} pairs.
- **Multi-turn Dialogue (439)** – LangChain scripts role-play *User*/*Bot* exchanges.

The resulting 12439 examples fine-tune KoGPT2 and act as the reference policy.

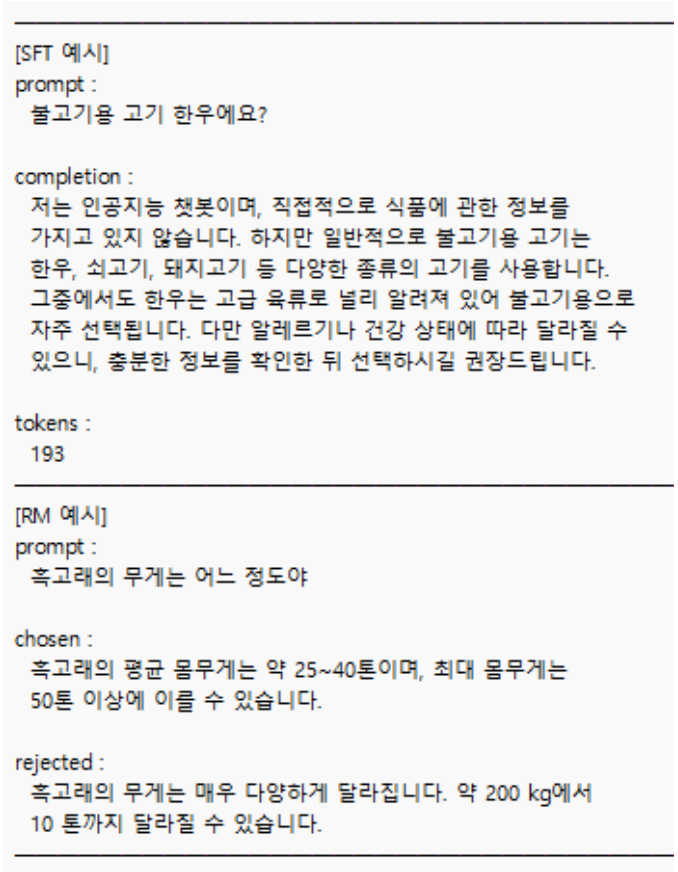

Fig. 1. Representative samples for SFT (top) and RM (bottom) datasets used to align KoGPT2.

C. Preference Pairs (used in both pipelines)

For 10220 prompts we generate three candidate answers (ChatGPT, GPT-3–`davinci`, GPT-3–`ada`) that human annotators rank (0–2). Each triplet becomes a binary {`prompt`, `chosen`, `rejected`} pair.

- **Pipeline A** – trains the reward model on these pairs.
- **Pipeline B** – feeds the same pairs directly to DPO.

D. Unlabelled Interaction Prompts (Pipeline A only)

PPO reuses the 12000 harvested prompts without labels; the trained reward model supplies on-the-fly feedback.

TABLE I
DATASET ARTEFACTS REQUIRED BY EACH PIPELINE

Dataset	Pipeline A	Pipeline B
SFT pairs (12 439)	✓	✓
Preference pairs (10 220)	✓	✓
Interaction prompts (12 000)	✓	—

III. METHODOLOGY

Starting from the same SFT KoGPT2 checkpoint, we branch into two alignment routes:

A. Supervised Fine-Tuning

KoGPT2-base is trained with token-level cross-entropy on the 12 878 SFT examples for 10 epochs (effective batch 8, FP16). The resulting model is also frozen as the reference policy.

B. Pipeline A: SFT–Reward Model–PPO

Reward Model. A scalar value head predicts reward r for `[Prompt ⊕ Response]`, trained via Eq. (**??**). **PPO.** Actor, critic, reward model, and reference form a four-network loop; the actor is optimised with a clipped objective plus KL penalty.

C. Pipeline B: SFT–Direct Preference Optimization

DPO skips reward estimation: for each preference triple it minimises the temperature-scaled loss of Eq. (**??**). Only the actor is updated, cutting memory and compute.

Why Direct Preference Optimisation (DPO)?: Direct Preference Optimisation (DPO) eliminates the reward–model/critic feedback loop by *directly* maximising the likelihood that the policy π_θ assigns higher probability to the preferred answer $\mathbf{a}^\star$ over a rejected answer $\tilde{\mathbf{a}}$ for the same prompt $\mathbf{p}$. Concretely, let

$$\Delta \log \pi_\theta = \log \pi_\theta(\mathbf{a}^\star \,|\, \mathbf{p}) - \log \pi_\theta(\tilde{\mathbf{a}} \,|\, \mathbf{p}),$$

then the DPO loss with temperature τ is

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}[\log \sigma(\tfrac{1}{\tau} \Delta \log \pi_\theta)],$$

where σ is the sigmoid. Thus the model is aligned *without* an explicit scalar reward, cutting memory and compute.

D. Training Hyper-parameters

TABLE II
KEY HYPER-PARAMETERS (ALL STAGES USE 10 EPOCHS)

Stage	Batch*	GradAcc	LR	Optimiser
SFT	2	4	5×10^{-5}	AdamW
RM	4	—	5×10^{-5}	Adam
PPO	8	—	1×10^{-5}	Adam
DPO	2	2	5×10^{-5}	Adam

* per-device training batch size.

a) Training time:

Stage	Wall-clock
SFT	1 h 40 m 32 s
RM	44 m 27 s
PPO	6 m 24 s
DPO	55 m 05 s
Total per pipeline	
SFT → RM → PPO	2 h 31 m 23 s
SFT → DPO	2 h 35 m 37 s

E. Metrics

We report automatic reference–based metrics that are widely adopted in dialogue and summarisation research:

a) ROUGE-2 (R-2).: ROUGE-n measures the n-gram overlap between the system output $\hat{y}$ and a reference y. For $n = 2$, it captures bigram co-occurrence, rewarding local phrase-level matches. We compute R-2 F_1 (harmonic mean of precision and recall) following the official `py-rouge` settings.

b) ROUGE-L (R-L).: ROUGE-L is based on the length of the *Longest Common Subsequence* (LCS) between $\hat{y}$ and y. Unlike R-2, it tolerates non-contiguous matches and therefore reflects sentence-level fluency and ordering. We again report the F_1 variant.

TABLE III
ROUGE PERFORMANCE OF ALIGNED KOGPT2 MODELS (HIGHER IS BETTER).

Model / Stage	**ROUGE-L**	**ROUGE-2**
SFT (baseline)	0.0091	0.00060
SFT → RM → PPO	0.0042	0.00027
SFT → DPO	0.0040	0.00023

IV. CONCLUSION

Our study compared two alignment strategies for KoGPT2 using the same set of preference-labeled data but diverging in method and system complexity: the full RLHF stack with PPO, and the simpler Direct Preference Optimization (DPO).

Quantitative findings.: Evaluation using ROUGE scores indicates that the baseline SFT model achieves the highest performance (ROUGE-L = 0.0091, ROUGE-2 = 0.00060). In contrast, both PPO and DPO variants yield lower scores, with PPO scoring ROUGE-L = 0.0042 and ROUGE-2 = 0.00027, and DPO at ROUGE-L = 0.0040 and ROUGE-2 = 0.00023. This suggests that neither preference-based method improves over the SFT baseline under automatic evaluation metrics, and DPO performs on par with PPO.

Qualitative analysis.: Manual inspection reveals limitations in both PPO and DPO generations. PPO occasionally outputs empty strings or English–Korean mixed text, likely due to overfitting on the reward model and poor KL penalty calibration. DPO avoids empty outputs but tends to generate irrelevant English phrases or excessively verbose responses. This implies that stylistic biases in the human-labeled preference data directly influence the actor in DPO, while PPO's multiple components may introduce their own inconsistencies.

Overall insight.: While DPO simplifies the alignment process by using a single model without auxiliary networks, it achieves results comparable to PPO in terms of output quality. However, both methods underperform relative to the initial SFT model. This highlights the importance of data quality in preference labels, and suggests future work should focus on improving annotation consistency and exploring hybrid approaches that combine DPO's simplicity with light reward shaping.

Limitations and Contribution.: A key limitation of our work lies in the insufficient effectiveness of the reward model training. In particular, the underperformance of both PPO and DPO relative to SFT may stem from suboptimal learning on the reward modeling data. This raises concerns about the reliability of the comparison, as the downstream models may not have fully benefited from preference supervision.

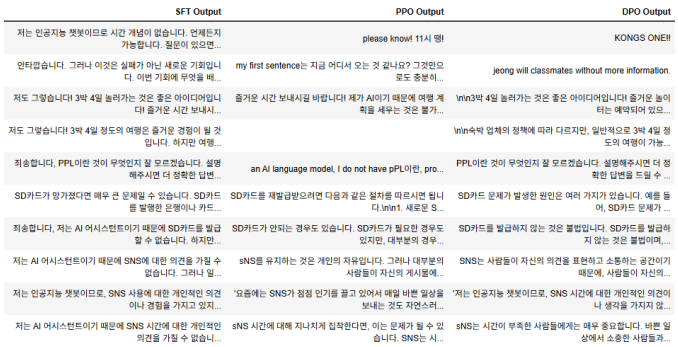

Fig. 2. Side-by-side output comparison from SFT, PPO, and DPO models on identical prompts. While SFT responses are more factually anchored, PPO and DPO sometimes hallucinate or respond generically.

Nonetheless, our study makes a meaningful contribution by applying Direct Preference Optimization (DPO) to a Korean language setting for the first time. Through this experiment, we demonstrate the feasibility of user preference-based alignment in non-English contexts and provide a foundation for future work on robust Korean RLHF pipelines.

Overall conclusion.: Under the same data budget, DPO delivers *comparable* ROUGE to PPO while collapsing the four-model RLHF stack into a single trainable policy, thereby eliminating auxiliary networks, reducing hyper-parameter surface area, and simplifying deployment. Nevertheless, the fact that both pipelines underperform the SFT baseline on ROUGE highlights that preference data quality and label balance—not merely algorithmic elegance—remain critical. Future work will (i) audit and re-label noisy preference pairs, (ii) perform grid searches over DPO temperature τ and PPO KL-penalty β, and (iii) explore hybrid strategies that combine DPO's structural simplicity with lightweight reward shaping to close the gap with the SFT baseline.